Entwicklung chemometrischer Methoden für die Klassifikation von Bakterien mittels Mikro-Raman-Spektroskopie

Von der Fakultät für Lebenswissenschaften
der Technischen Universität Carolo-Wilhelmina
zu Braunschweig
zur Erlangung des Grades einer
Doktorin der Naturwissenschaften
(Dr. rer. nat.)
genehmigte

D i s s e r t a t i o n

von Ulrike Schmid
aus Krumbach (Schwaben)

1. Referent: Professor Dr. Knut Baumann
2. Referent: Professor Dr. Hermann Wätzig
eingereicht am: 20.05.2009
mündliche Prüfung (Disputation) am: 06.07.2009

Druckjahr 2009

Meinen Eltern

Vorveröffentlichungen der Dissertation

Teilergebnisse aus dieser Arbeit wurden mit Genehmigung der Fakultät für Lebenswissenschaften, vertreten durch den Mentor der Arbeit, in folgenden Beiträgen vorab veröffentlicht:

Publikationen

U. Schmid, P. Rösch, M. Krause, M. Harz, J. Popp, K. Baumann, Gaussian Mixture Discriminant Analysis for the Single-Cell Differentiation of Bacteria Using Micro-Raman Spectroscopy, Chemometrics and Intelligent Laboratory Systems, 96 (2009), S. 159-171.

U. Neugebauer , U. Schmid , K. Baumann , H. Simon , M. Schmitt, J. Popp, DNA Tertiary Structure and Changes in DNA Supercoiling upon Interaction with Ethidium Bromide and Gyrase Monitored by UV Resonance Raman Spectroscopy, Journal of Raman Spectroscopy, 38 (2007), S. 1246-1258.

U. Neugebauer, U. Schmid, K. Baumann, W. Ziebuhr, S. Kozitskaya, U. Holzgrabe, M. Schmitt, J. Popp, The Influence of Fluoroquinolone Drugs on the Bacterial Growth of S. Epidermidis Utilizing the Unique Potential of Vibrational Spectroscopy, Journal of Physical Chemistry A; 111 (2007), S. 2898-2906

U. Neugebauer, U. Schmid, K. Baumann, W. Ziebuhr, S. Kozitskaya, V. Deckert, M. Schmitt, J. Popp, Towards a Detailed Understanding of Bacterial Metabolism-Spectroscopic Characterization of Staphylococcus Epidermidis, ChemPhysChem , 8 (2007), S. 124-137.

U. Neugebauer, U. Schmid, K. Baumann, U. Holzgrabe, W. Ziebuhr, S. Kozitskaya, W. Kiefer, M. Schmitt, J. Popp, Characterization of Bacterial Growth and the Influence of Antibiotics by Means of UV Resonance Raman Spectroscopy, Biopolymers, 82 (2006), S. 306-311.

Tagungsbeiträge

U. Schmid, K. Gaus, P. Rösch, J. Popp, K. Baumann, Rapid Identification of Bacteria Using Vibrational Spectroscopy – The Potential of Chemometric Tools (Poster), DPhG-Tagung, Mainz, 5-8 Oktober 2005.

U. Schmid, K. Gaus, P. Rösch, J. Popp, K. Baumann, Identification of Lactic Acid Bacteria Based on Vibrational Spectroscopy and Multivariate Statistics (Poster), SFB-Symposium, Würzburg, Februar 2006.

U. Schmid, P. Rösch, M. Harz, J. Popp, K. Baumann, Effectiveness of Chemometric Tools in Discrimination of Bacteria Based on Raman Spectroscopy (Poster), 4th International Chemometrics Research Meeting (ICRM), Veldhoven, Holland, 28. Mai-1. Juni 2006.

Inhaltsverzeichnis

Abkürzungen und Symbole

A_{max}	Maximaler Rang einer Matrix
a	Rang einer Matrix, alternativ: Grad eines Polynoms
ANOVA	Analysis of Variance (Varianzanalyse)
1.ABL	Raman-Spektren nach Basislinienkorrektur durch Bildung der 1. Ableitung
b	Basislinie eines Raman-Spektrums (Vektor mit Raman-Intensitäten)
b	Parameter, der die Verschiebung einer Hyperebene (SVMs) beschreibt
B	Diagonalmatrix mit Regressionskoeffizienten bei der PLS-Regression
BMU	Best Matching Unit (Gewinnerneuron beim Training einer SOM)
c	Klassenzugehörigkeit bei der Berechnung der ECE
c_{jr}	Zugehörigkeit eines oder mehrerer Objekte zu Klasse *j* und Subzentrum *r* in der MDA
c	Gewichtete **Y**-"Loadings" bei der PLS-Regression
C	Matrix gewichteter **Y**-"Loadings" bei der PLS-Regression
C	„Tuning"-Parameter bei der SVM-Klassifikation, der den erlaubten Grad an Missklassifikationen im Trainingsdatensatz reguliert; alternativ: Anzahl bekannter Klassen bei der Berechnung der ECE
C_j	Index, für die Zugehörigkeit eines oder mehrerer Objekte zu Klasse *j*
CV	Cross-Validation (Kreuzvalidierung)
CVANOVA	Cross-Validated Analysis of Variance (kreuzvalidierte Varianzanalyse)
$D(\mathbf{x},\mathbf{y})$	Distanz zwischen den Vektoren **x** und **y**
$d(x)$	Diskriminanzfunktion
df	Degrees of Freedom (Freiheitsgrade)
DNA	Desoxyribonukleinsäure
e	Spektrum nach Basisliniensubtraktion
e	Eulersche Zahl, e = 2,718281828459…
E	Fehlermatrix in der PLS-Regression

ECE	Empirical Conditional Entropie (empirische bedingte Entropie, siehe auch $H(c\|r)$)
ED(**x**,**y**)	Euklidische Distanz zwischen zwei Vektoren **x** und **y**
f	Lichtfrequenz
F	Fehlermatrix in der PLS-Regression
FDA	Food and Drug Administration
g_i	Klassenzugehörigkeit des Objektes *i*
gmf_j	Gauss´sche Mischfunktion für Klasse *j*
G	Schwingungszustand eines Moleküls
GMP	Good Manufacturing Practices (Gute Herstellungspraxis)
h	Planck´sche Konstante
h	Wellenzahl
H	Vandermonde Matrix der Wellenzahlen
$H(c\|r)$	Empirische bedingte Entropie (ECE) zwischen bekannten Klassen *c* und Cluster *r*
H	Geschätzte Varianz der Gruppen-Rangsummen beim Kruskal-Wallis-Test
HA1	Hauptachse 1
HA2	Hauptachse 2
$\mathbf{I_m}$	Einheitsmatrix mit der Dimension *m* x *m*
INTERPOL	Raman-Spektren nach Interpolation
IR	Infrarot
ISO	International Standards Organization
KL	Kullback-Leibler Distanz
k	Anzahl der Klassen (hier Bakterienstämme), alternativ: Anzahl der Gruppen bei einer ANOVA, alternativ: Anzahl nächster Nachbarn beim *k*-Nächste Nachbarn Klassifizierer
*k*NN	*k*-Nächste Nachbarn Klassifizierer
L^{mix}	Likelihood-Funktion
ℓ^{mix}	Logarithmierte Likelihood-Funktion
LDA	Lineare Diskriminanzanalyse

LOO-CV	„Leave-One-Out“ Kreuzvalidierung
LMO-CV	„Leave-Multiple-Out“ Kreuzvalidierung
M	Anzahl Stufen für Faktor A bei einer zweifaktoriellen ANOVA
m	Anzahl Variablen bzw. Dimensionen
MAD	Median Absolute Deviation (Median der absoluten Abweichungen)
MD(**x**,**y**)	Mahalanobis Distanz zwischen den Vektoren **x** und **y**
MDA	„Gaussian Mixture“ Diskriminanzanalyse
MLR	Multivariate lineare Regression
MS	Mean Squares (mittlere Abweichungsquadrate)
n	Anzahl Objekte
n_j	Anzahl Objekte, die zu Klasse j gehören
N	Anzahl Stufen für Faktor B bei einer zweifaktoriellen ANOVA, alternativ: Nachbarschaftsradius beim Training einer SOM
NIPALS	Nonlinear Iterative Partial Least Squares
p	Vektor der Länge k (Anzahl der Klassen) mit *a posteriori* Wahrscheinlichkeiten für jede Klasse
p_j	*a posteriori* Wahrscheinlichkeit eines Objektes bezüglich Klasse j
$p(x\|C_j)$	Klassenbedingte Wahrscheinlichkeitsdichte für Klasse j
$p(x)$	unbedingte Wahrscheinlichkeitsdichte
P	Matrix gewichteter **X**-“Loadings“ in der PLS-Regression
PCs	Principal Components (Hauptkomponenten)
PCA	Principal Component Analysis (Hauptkomponentenanalyse)
PCR	Principal Component Regression (Hauptkomponentenregression), alternativ: Polymerase Chain Reaction (Polymerasekettenreaktion)
PK-LDA	Paarweise LDA („major voting“)
PK-MDA	Paarweise MDA („major voting“)
PKPD	Algorithmus nach Price, Knerr, Personnaz und Dreyfus zur Bildung von Multi-Klassen *a posteriori* Wahrscheinlichkeiten nach Binarisierung
PLS	Partial Least Squares
PLS-DA	„Partial Least Squares“ Diskriminanzanalyse

POLY4	Raman-Spektren nach Basislinienkorrektur mit Robustem Polynomfit 4. Ordnung
$Pr(C_j)$	Wahrscheinlichkeit für die Zugehörigkeit eines beliebigen Objektes zur Klasse *j* (entspricht der *a priori* Wahrscheinlichkeit Π_j)
$Pr(C_j\|\mathbf{x})$	*a posteriori* Wahrscheinlichkeit, für die Zugehörigkeit eines Objektes **x** zur Klasse *j*
Q	**Y**-"Loadings" in der PLS-Regression
QDA	Quadratische Diskriminanzanalyse
r_i	Residuum des Datenpunktes *i* („Spike"-Eliminierung)
r	Residuum, alternativ: Laufindex für Subzentren bzw. Cluster in der MDA
$\tilde{r}_{0.5}$	Median aller Residuen („Spike"-Eliminierung)
R	Maß für die Glattheit einer Kurve (Whittaker-Algorithmus)
R	Anzahl Subzentren in der MDA bzw. Anzahl der Cluster bei Berechnung der ECE
RSS	Residual Sum of Squares (Summe der Abweichungsquadrate in der Regression)
s	„Cut-Off"-Wert (Grenzwert)
S	Diagonalmatrix mit Singulärwerten (Singulärwertzerlegung)
SERS	Surface-Enhanced Raman Spectroscopy
sgn	Vorzeichen
SS	Sum of Squares (Summe der Abweichungsquadrate)
SS_A	Summe der Abweichungsquadrate, die bei einer zweifaktoriellen ANOVA dem Einfluss des Faktors A zugeschrieben werden
SS_B	Summe der Abweichungsquadrate, die bei einer zweifaktoriellen ANOVA dem Einfluss des Faktors B zugeschrieben werden
SS_{AxB}	Summe der Abweichungsquadrate, die bei einer zweifaktoriellen ANOVA den Interaktionen zwischen Faktor A und B zugeschrieben werden
$SS_{BETWEEN}$	Summe der Abweichungsquadrate zwischen den Gruppen einer ANOVA
SS_{TOTAL}	Summe aller Abweichungsquadrate bei einer ANOVA

SS_{WITHIN}	Summe der Abweichungsquadrate innerhalb der Gruppen einer ANOVA
SOMs	Self Organizing Maps (Kohonenkarten)
SVD	Singular Value Decomposition (Singulärwertzerlegung)
SPIKEELIM	Raman-Spektren nach „Spike“-Eliminierung
SVMs	Support Vector Machines
t	„Scores“, alternativ: Anzahl der durchgeführten Trainingsschritte bei dem Training einer SOM
$\mathbf{T}$	„Score“-Matrix
TERS	Tip-Enhanced Raman Spectroscopy
u	Neuron
$\mathbf{U}$	Matrix der **Y**-“Scores“ in der PLS-Regression, alternativ: orthonormale Matrix in der Singulärwertzerlegung
$\tilde{\nu}$	Wellenzahl
$\mathbf{v}$	Gewichtsvektor eines Neurons beim Training einer SOM
$\mathbf{V}$	Orthonormale „Loadings“-Matrix (Hauptkomponentenanalyse)
$\mathbf{w}$	Normalenvektor der Hypperebene (SVMs), alternativ: gewichtete **X**-„Loadings“ in der PLS-Regression
$\mathbf{W}$	Matrix mit gewichteten **X**-“Loadings“ in der PLS-Regression
WHIT	Raman-Spektren nach Basislinienkorrektur mittels Whittaker-Algorithmus
Wu	Algorithmus nach Wu et al. zur Bildung von Multi-Klassen *a posteriori* Wahrscheinlichkeiten nach Binarisierung
z	Irrtumswahrscheinlichkeit (Ausreißererkennung)
α	Irrtumswahrscheinlichkeit
$\alpha(t)$	Lernrate als Funktion der Anzahl der Trainingsschritte t während des Trainings einer SOM
β	Regressionskoeffizient
$\boldsymbol{\beta}$	Vektor mit Regressionskoeffizienten
$\beta(u_{ij}, BMU, t)$	Nachbarschaftsfunktion während des Trainings einer SOM

δ_{ij}	geschätzte *a posteriori* Wahrscheinlichkeit für die Zugehörigkeit eines Objektes zu Klasse *i* bei der paarweisen Klassifikation zwischen Klasse *i* und *j* (siehe ω_{ij})
$\boldsymbol{\varepsilon}$	Vektor mit Abweichungen der Messpunkte von den Funktionswerten (Residuen) in der Regression
$\boldsymbol{\Sigma}$	Varianz-Kovarianz Matrix
$\lvert\boldsymbol{\Sigma}\rvert$	Determinante der Varianz-Kovarianz Matrix
λ	Wellenlänge
$\boldsymbol{\mu}_{jr}$,	Mittelwertsspektrum bezüglich Klasse *j* und Subzentrum *r* für die MDA
σ	Standardabweichung
π_{jr}	*a priori* Mischungs-Wahrscheinlichkeit bezüglich Klasse *j* und Subzentrum r bei der MDA
ρ	Parameter bei der Ein-Klassen-SVM; alternativ: Spearman´s Korrelationskoeffizient
τ	Fensterbreite beim Savitzky-Golay Algorithmus
ν	Anzahl der Durchläufe in der ν-fachen Kreuzvalidierung; alternativ: „Tuning“-Parameter in der Ein-Klassen-SVM, der die Irrtumswahrscheinlichkeit bei der Erkennung von Vorhersageausreißern angibt.
θ	Benutzerdefinierter Parameter beim asymmetrischen Whittaker-Smoother
ω_{ij}	„wahre“ *a posteriori* Wahrscheinlichkeit für die Zugehörigkeit eines Objektes zu Klasse *i* bei der paarweisen Klassifikation zwischen Klasse *i* und *j* (siehe δ_{ij})
ξ	Missklassifikationen für den Trainingsdatensatz in der SVM-Klassifikation
$\varphi(x)$	Kostenfunktion
Π_j	*a priori* Wahrscheinlichkeit für Klasse *j*
$\Re$	Menge reeler Zahlen

Mathematische Notation

In dieser Arbeit werden Skalare in kursiven Kleinbuchstaben (x) dargestellt, Vektoren in fetten Kleinbuchstaben ($\mathbf{x}$) und Matrizen in fetten Großbuchstaben ($\mathbf{X}$). Vektoren werden stets als Spaltenvektoren angegeben. Die transponierte Form eines Vektors $\mathbf{x}$ wird mit $\mathbf{x}^T$ bezeichnet. Gleiches gilt für transponierte Matrizen ($\mathbf{X}^T$). Die Dimension einer Matrix wird durch die Anzahl n ihrer Zeilen und die Anzahl m ihrer Spalten charakterisiert. Die i-te Spalte der Matrix $\mathbf{X}$ ist der Vektor $\mathbf{x}_i$.

Geschätzte Parameter werden mit einem Hütchen gekennzeichnet. So wird eine Schätzung des Vektors $\mathbf{x}$ als $\hat{\mathbf{x}}$ dargestellt. Die euklidische Norm eines Vektors $\mathbf{x}$ ist durch $\|\mathbf{x}\|$ gekennzeichnet. Der Mittelwert über alle Elemente von $\mathbf{x}$ heißt $\bar{x}$; entsprechend ergibt die Gesamtheit aller Spaltenmittelwerte einer Matrix $\mathbf{X}$ den Vektor $\bar{\mathbf{x}}$.

1 Einleitung

Bakterien spielen eine wichtige Rolle in unserem Leben. Sie umgeben uns immer und überall. Auf den menschlichen Organismus können sie sich sowohl nützlich als auch schädlich auswirken [1]. So übernimmt die Darmflora, die bei einem gesunden Menschen viele Billionen gutartiger Keime und Bakterien enthält, zahlreiche nützliche Funktionen wie den Schutz vor Krankheitserregern, die Anregung der Darmperistaltik, die Versorgung des Körpers mit Vitaminen und die Unterstützung der Verdauung. Eine Störung der Darmflora zieht unweigerlich eine Reihe von Gesundheitsproblemen nach sich. Auch die menschliche Hautflora besteht überwiegend aus Mikroorganismen. Sie spielt eine wichtige Rolle beim Schutz der Haut und des gesamten Organismus vor pathogenen Keimen. Bei zu häufigem und intensivem Waschen können Lücken in der Hautflora entstehen und der pH-Wert kann sich ins Basische verschieben. Dieses Milieu ist ideal für schädliche Bakterien wie beispielsweise *Staphylococcus aureus*. Über kleine Wunden können diese Keime in die Haut eindringen und sich dort vermehren, was zu einer Entzündung und zur Eiterbildung führen kann.

Als sehr problematisch erweisen sich Bakterien häufig in Kliniken (Erreger von Infektionskrankheiten, Bildung von Resistenzen), in der Lebensmittelindustrie (beschleunigter Verderb von Lebensmitteln) oder in der pharmazeutischen Herstellung (mögliche Kontamination pharmazeutischer Produkte).

In verschiedenen industriellen Herstellungsprozessen können mikrobielle Kontaminationen die Qualität der Produkte beeinträchtigen. Für die pharmazeutische Herstellung gibt es deshalb zahlreiche Richtlinien, die eine regelmäßige Kontrolle der Produktionsumgebung fordern [2]. So ist gemäß der guten Herstellungspraxis für medizinische Produkte (engl. Good Manufacturing Practices: GMPs) die routinemäßige Hygienekontrolle ein wichtiger Bestandteil des Produktionsablaufes [3]. Auch in der Lebensmittelherstellung muss laut der Verordnung 853/2004 des Europäischen Parlaments für jeden einzelnen Prozess nachgewiesen werden, dass die geforderten Hygienestandards erfüllt sind [4]. Besonders hohe Anforderungen werden an Produktionsbereiche gestellt, die zur Herstellung steriler

Produkte dienen (Reinräume). Entsprechende Richtlinien sind im Anhang 1 des EU-GMP-Leitfadens enthalten. Darin findet man die Einteilung in vier Reinraumklassen (A-D). Neben spezifischen Grenzwerten für die erlaubte Partikelkonzentration verschiedener Partikelbezugsgrößen sind dort auch Grenzwerte für die mikrobiologische Kontamination festgelegt. Die EN-ISO Normen der Reinraumtechnik bilden zudem eine gute Grundlage zur Umsetzung der von GMP geforderten Reinraum-Standards (EN ISO 14644 für die Reinraumtechnik allgemein und EN ISO 14698 für die Beherrschung der Biokontamination) [5]. Vor diesem Hintergrund haben die pharmazeutische Industrie sowie die Lebensmittelindustrie ein großes Interesse an einer schnellen und zuverlässigen Methode zur Identifizierung mikrobieller Kontamination während des Herstellungsprozesses in industriellen Reinräumen („Online-Monitoring“). Eine sofortige Analyse kann zu einer deutlichen Einsparung an Zeit und Produktionskosten führen. Die Identifizierung von Bakterien in mikrobiologischen Laboratorien kann allerdings mehrere Tage in Anspruch nehmen. Aus diesem Grund wurden in den letzten Jahrzehnten zahlreiche Forschungsvorhaben initiiert, die herkömmliche, mikrobiologische Methoden durch schnellere, leistungsfähigere Techniken ersetzen sollten. Neben verschiedenen neuen Verfahren wie Massenspektroskopie [6], Polymerase-Ketten-Reaktion (PCR) [7], Durchflusszytometrie [8] und Fluoreszenz-Spektroskopie [8], haben sich vor allem schwingungsspektroskopische Techniken als vielversprechend erwiesen [9,10]. Sowohl mittels IR- [11,12] als auch mittels Raman-Spektroskopie [13-17] können Mikroorganismen zuverlässig identifiziert werden. Bei der IR-Spektroskopie müssen die Bakterienzellen vor der Analyse kultiviert werden. Um eine zuverlässige Analyse zu gewährleisten, läuft die Kultivierung unter standardisierten Bedingungen ab. Das heißt, dass Kultivierungsmedium, Kultivierungsdauer und Temperatur streng kontrolliert werden [18]. Da Wasser die Aufnahme von IR-Spektren stört, werden die Proben anschließend getrocknet [19]. Während die eigentliche IR-spektroskopische Messung nur wenige Minuten dauert, nimmt die aufwändige Probenvorbereitung mehrere Stunden in Anspruch.

Im Gegensatz zur IR-Spektroskopie ist die Probenvorbereitung in der Raman-Spektroskopie sehr einfach. Wasser ist unproblematisch für Raman-spektroskopische Messungen. Außerdem können durch die Erfindung der konfokalen Mikro-Raman-Spektroskopie sogar einzelne Bakterienzellen innerhalb weniger Minuten und ohne vorherige Kultivierung

vermessen werden [14,20]. Dies eröffnet neue experimentelle aber auch datenanalytische Herausforderungen. Letztere sind Gegenstand dieser Arbeit und werden im Folgenden näher ausgeführt. Die Raman-Strahlung kann grundsätzlich durch Bestrahlen mit Licht des UV-, IR- oder des sichtbaren Bereichs angeregt werden. Der IR-Bereich scheidet aufgrund der geringen Nachweisgrenze für die Analyse einzelner Bakterienzellen aus. Die UV-Resonanz-Raman-Spektroskopie bietet den Vorteil, dass die Raman-Strahlung nicht durch Fluoreszenz überlagert wird. Allerdings kann es während der Messung leicht zu einer photochemischen Zersetzung der Proben kommen. Dieses Risiko besteht nicht bei der Anregung im sichtbaren Bereich, weshalb diese Methode für eine schnelle und einfache Bestimmung von einzelnen Bakterienzellen gut geeignet ist. Um das Problem der auftretenden Fluoreszenz zu lösen, wurden spezielle Verfahren wie die „oberflächenverstärkte Raman-Spektroskopie“ (engl. Surface-Enhanced Raman Spectroscopy: SERS) [21-23] und die „spitzenverstärkte Raman-Spektroskopie" (engl. Tip-Enhanced Raman Spectroscopy: TERS) [24,25] zur Steigerung der Intensität der Raman-Strahlung entwickelt. In dieser Arbeit erfolgt die Anregung der Raman-Strahlung im sichtbaren Bereich bei 532 nm.

Für die spektroskopische Identifizierung von einzelnen Bakterienzellen müssen mehrere Punkte in der Datenauswertung berücksichtigt werden. In einigen Studien wurde gezeigt, dass Kultivierungsbedingungen und bakterielle Wachstumszustände die spektralen Eigenschaften stark beeinflussen [17]. Bei einer direkten Analyse der Bakterien im industriellen Umfeld sind die Wachstumsbedingungen aber nicht bekannt. Um die Reproduzierbarkeit der Klassifikationsergebnisse zu gewährleisten, muss die Wiedererkennung der Bakterienstämme unabhängig von diesen Parametern sein. In den letzten Jahren haben einige Forschergruppen den Einfluss verschiedener Wachstumsparameter auf die Identifizierung von einzelnen Bakterienzellen mittels Mikro-Raman-Spektroskopie -teils auf Stamm-Ebene, teils auf Art-Ebene- untersucht. Es stellte sich heraus, dass die für diesen Zweck größtenteils verwendeten datenanalytischen Methoden der unüberwachten Mustererkennung (z. B. hierarchische Clusteranalyse) in der Analyse heterogen zusammengesetzter bakterieller Datensätze nicht ausreichend waren [26-28]. Eine deutliche Trennung der Bakterienarten und –stämme konnte mit diesen Methoden nur erreicht werden, wenn die Bakterien vor den spektroskopischen Messungen unter standardisierten Kultivierungsbedingungen behandelt wurden. Daraufhin wurden komplexere

datenanalytische Methoden für diese Aufgabe vorgeschlagen. Hutsebaut et al. differenzierten erfolgreich die Kolonien von 30 unterschiedlich kultivierten *Bacillus* Stämmen auf Art-Ebene unter Verwendung der Linearen Diskriminanzanalyse (LDA) [26]. Xie et al. schlugen die Generalisierte Diskriminanzanalyse (GDA) vor, um sechs Bakterienarten zu differenzieren, die sich in verschiedenen Wachstumsphasen befanden [27]. Für die Klassifikation von hochdiversen Datensätzen bestehend aus Bakterien- und Hefezellen führten Rösch et al. „Support Vector Machines“ (SVMs) ein [14-17]. M. Harz et al. bestätigten das Potential von SVMs für die Analyse heterogener Bakterien-Datensätze, indem sie *Staphylococcus* Stämme, die unter verschiedensten Bedingungen kultiviert wurden, differenzierten [28]. Ein Nachteil von SVMs ist ihre Komplexität, die ihre Anwendbarkeit in bestimmten Situationen begrenzt. Im klinischen und industriellen Umfeld ist die Beurteilung der Zuverlässigkeit der getroffenen Vorhersagen ebenso von Bedeutung wie die Klassifikation der Bakterien. Daneben müssen sogenannte Vorhersageausreißer erkannt werden. Das sind neue Spektren, die von Mikroorganismen stammen, die nicht zu dem registrierten Datensatz gehören. Außerdem birgt eine bloße Klassifikation ohne die Berücksichtigung von Artefakten und störenden Einflüssen Risiken und Fehlerquellen. So kann die Datenstruktur (Ähnlichkeitsbeziehungen zwischen den Spektren) von Faktoren beeinflusst sein, die unabhängig von der Identität (Stammzugehörigkeit) der analysierten Bakterien sind. Dazu zählen beispielswiese Messartefakte sowie unterschiedliche Stoffwechselzustände und Wachstumsphasen von Bakterien, die sich in den Spektren niederschlagen und die Robustheit der Klassifikationsergebnisse beeinflussen. Um reproduzierbare Klassifikationsraten zu garantieren, muss vor allem der Effekt von Kultivierungsbedingungen auf die Datenstruktur und den Klassifikationserfolg untersucht werden. Aus diesem Grund sind einfache und leicht interpretierbare Modelle häufig von Vorteil. In den letzten Jahren wurde intensiv an der Weiterentwicklung der SVMs gearbeitet, so dass *a posteriori* Wahrscheinlichkeiten erhalten werden können, mit denen die Zuverlässigkeit der einzelnen Vorhersagen abgeschätzt werden kann [29]. Auch die Erkennung von Vorhersageausreißern ist möglich [30]. Aufgrund der hohen Komplexität der SVMs ist eine Interpretierbarkeit des Models allerdings nicht gegeben.

Das Ziel dieser Arbeit ist es, ein umfassendes Auswertungsverfahren für die Identifizierung einzelner Bakterienzellen mittels Mikro-Raman-Spektroskopie zu entwickeln. Das Verfahren

soll für ein „Online-Monitoring“ in der pharmazeutischen Reinraum-Herstellung geeignet sein. Die zu untersuchenden Bakterien sind Stämme, die bevorzugt in industriellen Reinräumen vorkommen. Als Grundlage für die Datenanalyse, muss zunächst eine geeignete Spektren-Vorbehandlung gefunden werden. Darauf aufbauend ist es das Ziel eine möglichst präzise Modellierung der Bakterienklassifikation zu erreichen. Für die Wahl der Klassifikationsmodelle ist die Vorhersagegüte entscheidend, aber auch die Einfachheit der Berechnung und die Interpretierbarkeit des Modells. So werden neben komplexen Verfahren wie SVMs leicht interpretierbare Modelle auf ihre Tauglichkeit für die gegebene Aufgabenstellung untersucht. Insbesondere wird der Einfluss verschiedener Kultivierungsbedingungen auf den Klassifikationserfolg analysiert. Um auch Bakterien zu erkennen, die unbekannt sind, d.h. Bakterien, die nicht im Trainingsdatensatz enthalten sind, wird eine zuverlässige Methode zur Erkennung von Vorhersage-Ausreißern entwickelt.

2 Theoretische Grundlagen

2.1 Die Bakterienzelle

Zum besseren Verständnis der in dieser Arbeit behandelten Raman-Spektren und der darin enthaltenen chemischen Information, wird im Folgenden der grundsätzliche Aufbau einer Bakterienzelle [31,32] beschrieben.

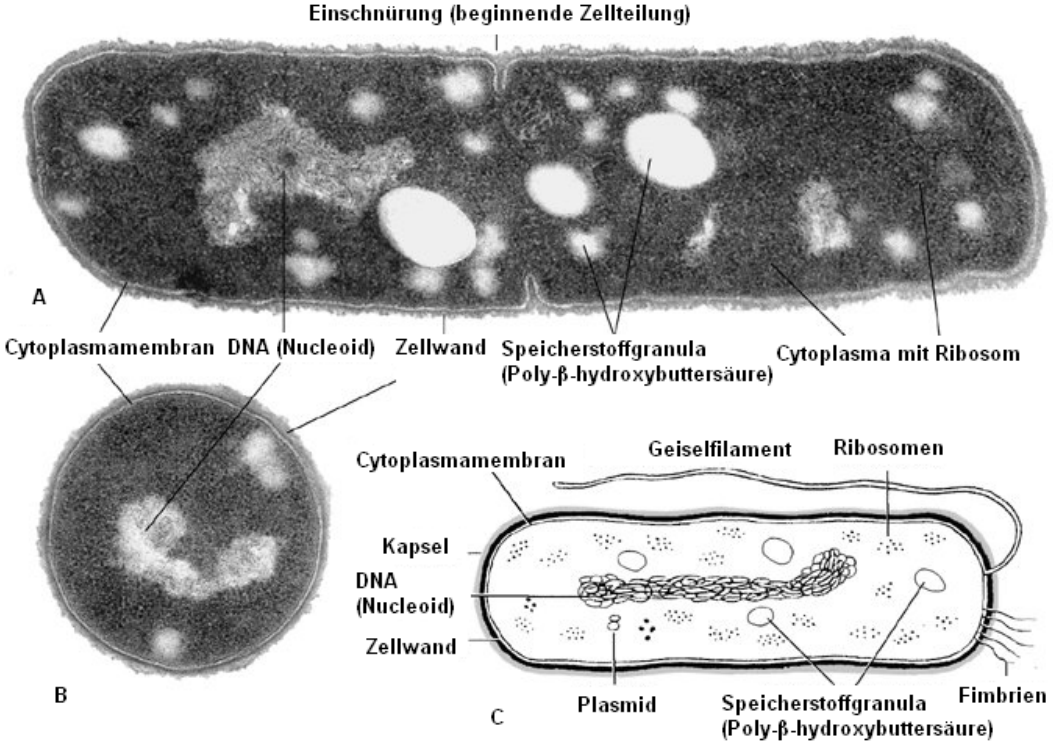


Abbildung 2.1: Aufbau einer Bakterienzelle: Elektronenmikroskopische Aufnahme des grampositiven Bakteriums *Bacillus subtilis* **A:** im Längsschnitt, **B:** im Querschnitt. **C:** Schematischer Aufbau einer begeißelten Bakterienzelle (gramnegatives Bakterium) (aus [32] mit freundlicher Genehmigung des Spektrum Verlags).

Bakterien sind prokaryotische, mikroskopisch kleine Lebewesen mit einem Durchmesser von ca. 1 µm. Sie zählen zu den Mikroorganismen. Die Bezeichnung prokaryotisch beschreibt die Tatsache, dass Bakterien keinen echten Zellkern besitzen (siehe Abbildung 2.1). Stattdessen liegt ein zirkuläres Chromosom aus Desoxyribonukleinsäure (DNA) frei im Cytoplasma vor. Dieses wird als Nukleoid (Kernäquivalent) bezeichnet. Zusätzlich können ein bis mehrere extrachromosomale, sich autonom replizierende genetische Elemente (Plasmide) vorhanden sein.

In ihrer Grundstruktur sind Bakterien von einer Cytoplasmamembran (Phospholipid-Doppelschicht) und in der Regel von einer Bakterienzellwand (Peptidoglykan) umgeben. Im Innern der Bakterien findet man zahlreiche Einschlüsse. Hierzu zählen unter anderem Speicher- und Reservestoffe wie Polysaccharide, Fette, Polyhydroxybuttersäure, Schwefeltropfen und Polyphosphatgranula sowie Gasvakuolen und Kristallkörper. Organellen wie Mitochondrien und Chloroplasten, die in eukaryotischen Zellen vorkommen, sind in der Bakterienzelle nicht vorhanden.

Ein wichtiges Unterscheidungskriterium ist die Einteilung in grampositive (z. B. *Streptokokken* und *Staphylokokken*) und gramnegative (z. B. *E. coli* und *Salmonella*) Bakterien. Die Zellwandzusammensetzungen beider Formen sind sich sehr ähnlich. Sie bestehen aus Peptidoglykanschichten, die auch als Murein bezeichnet werden. Dabei handelt es sich um eine netzartige Struktur von Polysaccharidketten, dem Glykan, die über kurze Peptidketten (meist Tetra-Peptide) miteinander verknüpft sind. An dem Aufbau des Glykans sind zwei Aminozucker beteiligt, das N-Acetylglucosamin und die N-Acetylmuraminsäure, die in alternierender Reihenfolge über β-1,4-Bindungen lange, unverzweigte Ketten bilden.

Die Zellwand der gramnegativen Bakterien ist wesentlich dünner als die der grampositiven Bakterien. Im Gegensatz zu den grampositiven Bakterien ist ihre Zellwand nach außen durch eine weitere Membran begrenzt, die neben einer inneren Phospholipidschicht an der Außenseite aus Lipopolysacchariden (LPS) besteht. Ein Charakteristikum für grampositive Bakterien sind große Mengen von Teichonsäuren, die kovalent im Mureinnetz verankert sind. Dabei handelt es sich um Ribitol-Phosphat-Polymere (Ribitol-Teichonsäuren) oder Glycerol-Phosphat-Polymere (Glycerol-Teichonsäuren), die kettenartig nach außen ragen.

Grampositive und gramnegative Bakterien lassen sich durch die sogenannte Gram-Färbung [33] unterscheiden. Dabei werden Bakterien auf einem Objektträger fixiert und mit

Kristallviolett gefärbt. Dadurch erhalten alle Bakterien eine blaue Farbe. Danach wird mit Jod-Kaliumjodid gebeizt und mit Alkohol entfärbt. Bei der Entfärbung verhalten sich die Bakterien unterschiedlich. Grampositive Bakterien behalten die blaue Farbe bei, während gramnegative Bakterien den Farbstoff wieder abgeben. Die gramnegativen Bakterien sind dann nur noch schlecht zu erkennen. Deshalb wendet man bei ihnen eine Gegenfärbung (basisches Fuchsin bzw. Safranin) an, so dass sie eine rote Farbe erhalten.

Viele Bakterien bewegen sich mit Hilfe rotierender Geißeln fort. Das in Abbildung 2.1 gezeigte Bakterium besitzt nur eine Geißel – andere, wie *E. coli,* haben zahlreiche Geißeln, die über die gesamte Zelle verteilt sind. Mit den deutlich kleineren Fimbrien dagegen können sich Bakterien an Oberflächen festheften. Sowohl Geißeln als auch Fimbrien sind über große Multiproteinkomplexe in der Zelloberfläche verankert.

Unter bestimmten Bedingungen bilden einige Gattungen (z. B. *Clostridium* und *Bacillus*) Sporen. Dies sind stoffwechselinaktive Dauer- und Ausbreitungsformen der Bakterien. Man unterscheidet Endosporen, Exosporen, Myxosporen und Cysten. Meist wird die Bildung der Dauerformen durch einen Mangel an Nährstoffen oder andere ungünstige Wachstumsbedingungen ausgelöst. Kommen Sporen in ein günstiges Umfeld, werden sie wieder stoffwechselaktiv und vermehren sich. Eine charakteristische Substanz, die im Kern aller Endosporen vorkommt, ist die Dipicolinsäure (Pyridin-2,6-dicarbonsäure), die häufig als Calciumchelatkomplex vorliegt. In vegetativen Zellen ist sie nicht enthalten.

Dieser Einblick in die Zellzusammensetzung von Bakterien zeigt, wie vielfältig die chemische Information ist, die bei der Raman-spektroskopischen Untersuchung von Bakterien eine Rolle spielt. Durch das hohe Empfindlichkeitsniveau der konfokalen Raman-Spektroskopie werden bereits kleine Unterschiede im Aufbau einzelner Bakterienzellen in den Spektren erkennbar, was die Raman-Spektroskopie zu einer idealen Methode für die Identifizierung von Bakterien macht. Im Folgenden werden die Grundlagen der Raman-Spektroskopie vorgestellt. Anschließend wird anhand einiger Mikro-Raman-Spektren gezeigt, wie bestimmte spektrale Bereiche verschiedenen funktionellen Gruppen in Bakterien zugeordnet werden können.

2.2 Raman-Spektroskopie

2.2.1 Theorie der Raman-Spektroskopie

Die Raman-Spektroskopie [34] gehört wie die Infrarot (IR)-Spektroskopie [35] zu den schwingungsspektroskopischen Methoden. Beide Techniken basieren darauf, dass Moleküle bei Einstrahlung von Licht bestimmter Wellenlänge Molekülschwingungen ausführen. Die Schwingungsfrequenz hängt von den im Molekül enthaltenen Atomtypen (bzw. deren Masse) und der Art der Bindungen (z. B. Einfach-, Zweifach-, Dreifachbindungen) ab. Je nach Schwingungsrichtung unterscheidet man Streckschwingungen (Valenzschwingungen), die entlang der Bindungsachse zweier Atome verlaufen, und Beugeschwingungen (Deformationsschwingungen), die unter Deformation des Bindungswinkels erfolgen. Beugeschwingungen treten immer bei tieferen Wellenzahlen (kleinere Energie) auf als Streckschwingungen. Ein Molekül kann bestimmte Schwingungsniveaus *G* annehmen. Diese sind schematisch in Abbildung 2.2 gezeigt. Die Änderungen des Schwingungszustandes sind mit Energieänderungen verbunden. Der typische Wellenzahlbereich, bei dem Moleküle in höhere Schwingungszustände übergehen können, liegt zwischen 400 und 4000 cm^{-1}, was dem Wellenlängenbereich 25 bis 2.5 µm entspricht (IR-Bereich).

Während in der IR-Spektroskopie nach Lichteinstrahlung der Anteil des direkt absorbierten Lichtes gemessen wird, werden in der Raman-Spektroskopie die inelastisch gestreuten Lichtanteile detektiert. Die inelastische Streuung, die auch als Raman-Effekt bezeichnet wird, wurde im Jahr 1928 von dem indischen Physiker Sir Chandrasekhara Venkata Raman experimentell nachgewiesen. Für diese Entdeckung wurde Sir Raman im Jahr 1930 mit dem Nobel-Preis ausgezeichnet.

Die Entstehung der Streueffekte lässt sich quantenmechanisch folgendermaßen erklären. Werden Moleküle mit monochromatischem Licht bestrahlt, tritt der größte Teil dieser Strahlung ungehindert durch die Probe hindurch (Transmission). Ein geringerer Teil der Strahlung wird an den Molekülen abgelenkt und in alle Raumrichtungen gestreut. Dabei kann die Streuung elastisch oder inelastisch erfolgen. Als Rayleigh-Streuung wird die elektromagnetische Strahlung bezeichnet, die elastisch, also ohne messbaren Energieverlust, gestreut wird. Die Frequenz der Rayleigh-Streuung und die Frequenz der

Anregungswellenlänge sind demzufolge gleich (siehe Abbildung 2.2). Ein sehr viel kleinerer Teil hingegen (ca. 1 von 10^7 Photonen) erfährt eine inelastische Wechselwirkung (inelastischer Stoß) mit den Molekülen. Man spricht von der Raman-Streuung oder dem Raman-Effekt. Die Energie, die bei der inelastischen Streuung abgegeben oder aufgenommen wird, entspricht der Differenz zwischen zwei Energieniveaus einer Molekülschwingung.

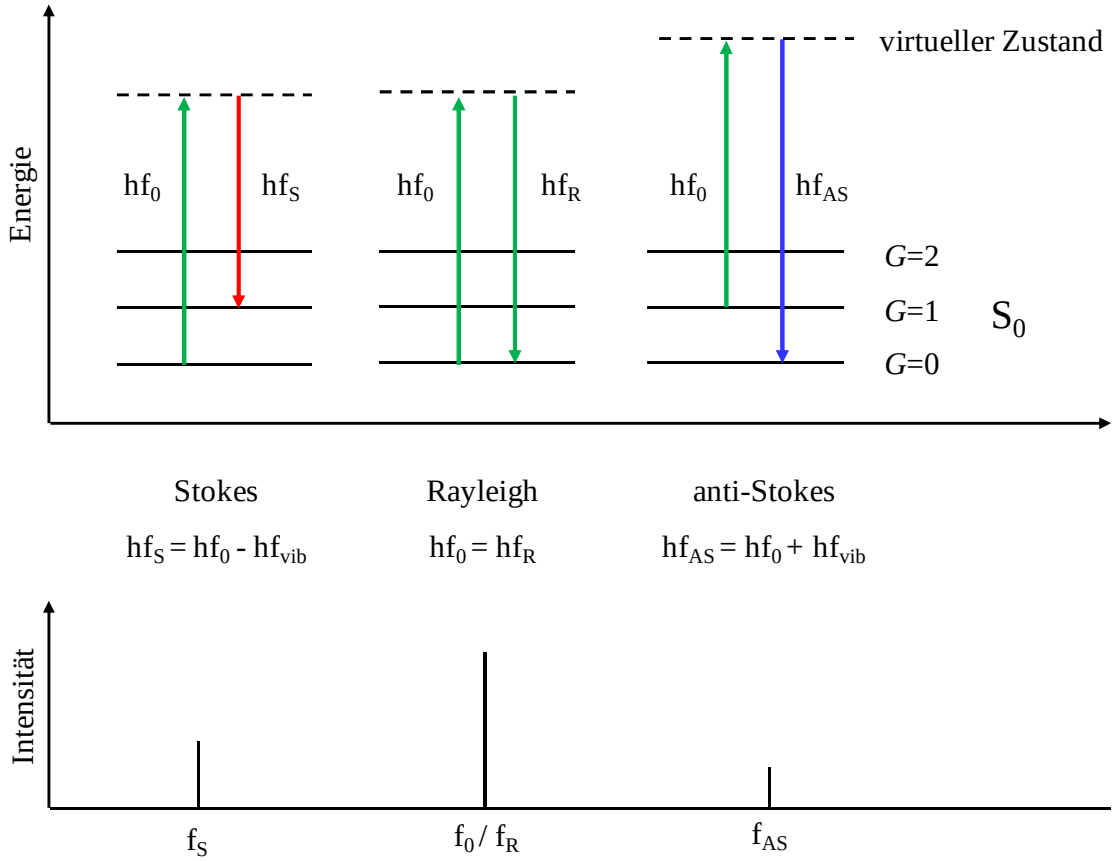


Abbildung 2.2: Schematische Darstellung der Energiezustände bei der Rayleigh- und Raman-Streuung (modifiziert nach [36]). Bei der Raman-Streuung kann die Wellenlänge des eingestrahlten Lichts (grüner Pfeil) entweder zum langwelligen Spektralbereich (Stokes-Raman-Streuung: roter Pfeil) oder zum kurzwelligen Spektralbereich (anti-Stokes-Raman-Streuung: blauer Pfeil) verschoben sein. S_0 bezeichnet den elektronischen Grundzustand; G=0 entspricht dem Schwingungsgrundzustand; G=1 und G=2 sind schwingungsangeregte Zustände. Die Lichtenergie wird berechnet als das Produkt aus Stahlungsfrequenz f des Lichts und der Planck Konstante h. So bezeichnen hf_0, hf_S, hf_R und hf_{AS} die jeweiligen Energiewerte für eingestrahltes Licht, Stokes-, Rayleigh- und anti-Stokes-Streuung. hf_{vib} entspricht der Energie des Schwingungszustandes G=1.

Es werden zwei Arten von inelastischer Streuung unterschieden:

- *Stokes-Raman-Streuung*: Nach Bestrahlung und inelastischer Streuung besitzt das Molekül eine höhere Schwingungsenergie als zuvor. Das Streulicht ist dabei energieärmer (siehe Abbildung 2.2: roter Pfeil) als das eingestrahlte Licht und weist eine niedrigere Frequenz auf.
- *anti-Stokes-Raman-Streuung*: Nach Bestrahlung und inelastischer Streuung besitzt das Molekül eine niedrigere Schwingungsenergie als zuvor. Das Streulicht ist energiereicher (siehe Abbildung 2.2: blauer Pfeil) als das eingestrahlte Licht und weist eine höhere Frequenz auf. Dieser Fall ist nur möglich, wenn sich das Molekül vor der Anregung in einem höheren Schwingungszustand als dem Schwingungsgrundzustand befindet.

Die Intensität der anti-Stokes-Streuung ist geringer als die der Stokes-Streuung, was darin begründet ist, dass bei ersterer vor dem inelastischen Stoß bereits Schwingungen vorhanden sein müssen und dies weniger häufig auftritt. Deshalb wird bei der Messung von Raman-Spektren normalerweise nur die Stokes-Raman-Streuung aufgenommen. Da die Streustrahlung nur weniger als 1% des eingestrahlten Lichts ausmacht, müssen sehr intensive Lichtquellen eingesetzt werden. In der Regel werden dafür Laser benutzt, die sich auch aufgrund ihres monochromatischen Lichts gut für die Messung eignen. Der Aufbau und die Funktionsweise des in der vorliegenden Dissertation verwendeten konfokalen Mikro-Raman-Spektrometers wird im experimentellen Teil der Arbeit beschrieben (siehe Kapitel 3.1).

2.2.2 Raman-Spektroskopie an Bakterien

Der Einsatz schwingungsspektroskopischer Messungen zur Identifizierung von Bakterien wurde erstmals in den 50er und 60er Jahren erforscht [37-39]. Zunächst wurden nur IR-spektroskopische Messungen durchgeführt, während die Raman-Spektroskopie aufgrund der zu dieser Zeit bestehenden Nachteile (z. B. höhere Kosten, geringere Geschwindigkeit, geringere Empfindlichkeit, höhere Komplexität) wenig Aufmerksamkeit erhielt. Erst mit den Entwicklungen in der Lasertechnologie nahm die Anwendung der Raman-Spektroskopie in

der Durchführung biologischer Studien zu. So erschienen in den 80er Jahren erste Publikationen zur Raman-spektroskopischen Differenzierung von Bakterien [40,41]. In den letzten Jahrzehnten konnten Empfindlichkeit, Reproduzierbarkeit und Handhabbarkeit aller spektroskopischen Techniken wesentlich verbessert werden, was ihre Praxisrelevanz für diesen Zweck deutlich erhöhte. Durch die Erfindung der konfokalen Mikro-Raman-Spektroskopie (siehe Kapitel 3.1.1) können sogar Messungen an einzelnen Bakterienzellen durchgeführt werden, wodurch eine vorherige Kultivierung der Bakterien entfallen kann. Schwingungsspektroskopische Methoden liefern nichtinvasiv einen sehr spezifischen "Fingerabdruck" von mikrobiellen Zellen, was sowohl zur Charakterisierung als auch zur Identifizierung von Bakterien genutzt werden kann. Abbildung 2.3 zeigt typische Mikro-Raman-Spektren von Bakterien bei Anregung im sichtbaren Wellenlängenbereich.

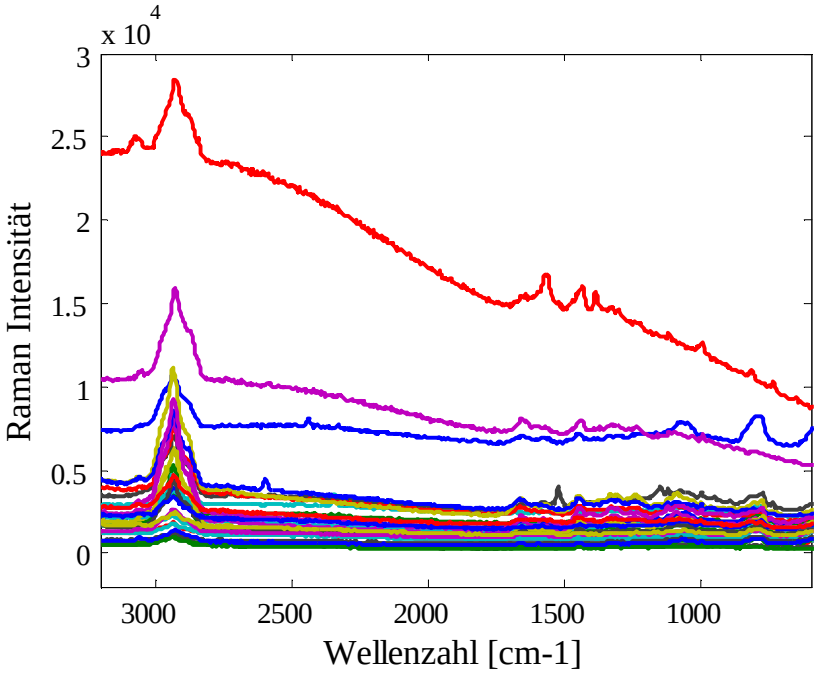


Abbildung 2.3: Mikro-Raman-Spektren von 29 Bakterien-Stämmen, die in industriellen Reinräumen vorkommen.

In den Mikro-Raman-Spektren wird die Raman-Intensität gegen die Wellenzahl aufgetragen. Die Wellenzahl $\tilde{\nu}$ bezeichnet in der Spektroskopie den Kehrwert der Wellenlänge λ: Es besteht folgender Zusammenhang:

$$\tilde{\nu} = \frac{\mathrm{f}}{\mathrm{c}} = \frac{1}{\lambda} \tag{2.1}$$

Dabei steht c für die Lichtgeschwindigkeit und f für die Lichtfrequenz.

Tabelle 2.1: Zuordnung von einigen Banden, die bei der Aufnahme von Raman-Spektren von Mikroorganismen häufig auftreten [9,13,40,41] .

Wellenzahl [cm^{-1}]	**Zuordnung**
3069	(C=C-H) aromatisch, olefinisch (s)
2937	CH_3 und CH_2 (s)
1660	Amid I
1614	Tyrosin
1607	Phenylalanin
1575	Guanin + Adenin Ring (s)
1450	CH_2 (def)
1242	Amid III
1128	C-N and C-C (s)
1092	CC-Gerüst, COC (s) von glykosidischen Bindungen
1001	aromatischer Ring (Phenylalanin)
899	COC (s)
855	CC (s), COC 1,4 glycosidische Bindung, "buried" Tyrosin
782	Ringstreckschwingung (Cytosin, Uracil)
621	Phenylalanin (Gerüst)

(s)=Streckschwingung, (def) =Deformationsschwingung

Aus Raman-Spektren erhält man Informationen über die allgemeine chemische Zusammensetzung der Bakterien. In Tabelle 2.1 ist die Zuordnung einiger wichtiger Banden beschrieben [9,13,40,41]. Besonders auffällig sind die CH-Steckschwingungsbanden im Bereich von 2700 bis 3000 cm^{-1}. Diese Banden lassen sich auf CH_3-, CH_2- und CH-funktionelle Gruppen von Lipiden, Proteinen und Kohlenhydraten zurückführen und sind somit ein Charakteristikum für das gesamte organische Material. Daneben findet man zahlreiche kleinere Peaks. Banden, die Proteinen zugeordnet werden können, sind die Amid-I-Bande bei 1660 cm^{-1} und die Amid-III-Bande in der Spektren-Region zwischen 1220 und 1300 cm^{-1}. Die Amid-I-Bande kommt hauptsächlich durch eine C=O−Streckschwingung zustande, während an der Amid-III-Bande vor allem die C−N−Streckschwingung gekoppelt mit einer N−H-Deformationsschwingung beteiligt ist. Weiterhin findet man die aromatische Ring-Steckschwingung von Phenylalanin um 1000 cm^{-1}. Die Banden um 1450 und 1333 cm^{-1} können CH_2- und CH-Beugeschwingungen zugeordnet werden. Nukleinsäure-Schwingungen findet man mit der Guanin- und Adenin-Ring-Steckschwingung bei 1575 cm^{-1}.
Die Zuordnung der funktionellen Gruppen zu den jeweiligen Banden der Raman-Spektren ermöglicht es, Aussagen über die Zellzusammensetzung von Bakterien zu treffen. Da verschiedene Bakterienstämme eine unterschiedliche biochemische Zusammensetzung aufweisen, die sich in den Spektren niederschlägt, ist es möglich, sie mit Hilfe der Raman-Spektren zu differenzieren. Die Unterschiede in den Spektren sind häufig so klein, dass man sie mit dem bloßen Auge nicht erkennen kann. Man benötigt deshalb eine computergestützte Auswertung. Dabei werden nicht nur einzelne Teile des Spektrums verwendet, sondern die Information des gesamten Spektrums kann mit Hilfe von Klassifikationsmethoden zur Differenzierung der Bakterien genutzt werden. Die Theorie der zu diesem Zweck verwendeten multivariaten Verfahren wird in den folgenden Kapiteln vorgestellt.

2.3 Multivariate Datenanalyse

In diesem Teil der Arbeit werden die mathematischen Grundlagen für die Auswertung der spektroskopischen Daten beschrieben. Die Anwendung statistischer und mathematischer Methoden auf chemisch experimentelle Daten wird als Chemometrik oder Chemometrie [42]

bezeichnet. Durch die großen Fortschritte im Bereich der instrumentellen Analytik sowie in der Computertechnik in den letzten Jahrzehnten sind die Probleme, die sich in der heutigen Zeit an diese Disziplin stellen, zunehmend vielschichtig und komplex. Dadurch wächst auch die Bandbreite und Komplexität der verwendeten mathematischen Methoden. Vor allem multivariate Techniken, die viele Faktoren gleichzeitig (hier das gesamte Spektrum) in die Berechnungen einbeziehen, sind heute in der Auswertung chemischer Daten unverzichtbar und stellen den Hauptteil dieser Arbeit dar.

2.3.1 Datenvorbehandlung

Die Datenvorbehandlung ist ein wichtiger Aspekt in der Auswertung von Raman-Spektren. Ziel der Vorbehandlung ist es, irrelevante oder zufällige Variationsquellen (Rauschen) und systematische Fehlerquellen in den Spektren zu reduzieren oder komplett zu entfernen. Dabei wird die spektrale Information verändert, was sich sowohl positiv als auch negativ auswirken kann. Ein Beispiel für Störeinflüsse in Raman-Spektren sind multiplikative Effekte, die beispielsweise durch unterschiedliche Schichtdicken der biologischen Proben entstehen können. Daneben kann mangelnde Gerätejustierung zu systematischen Verschiebungen in den Spektren führen. Die zwei Hauptgründe für Fehlsignale in Raman-Spektren sind jedoch die Sensitivität der CCD-Kamera (engl. Charge-Coupled Device Camera) gegenüber kosmischer Strahlung und die Eigenfluoreszenz der organischen Moleküle. Beides verursacht durch Überlagerung der Raman-Strahlung additive Effekte. Die CCD-Kamera, deren hohe Sensitivität für die Detektion der schwachen Raman-Strahlung ideal ist, reagiert auch auf kosmische Strahlung empfindlich. Dies führt zu scharfen Peaks („Spikes“) in den Spektren, die keine chemische Information enthalten (siehe Abbildung 2.5). Durch Eigenfluoreszenz der biologischen Proben entstehen dagegen horizontal versetzte oder langsam steigende bzw. fallende Basislinien. Aufgrund der unterschiedlichen Arten des Rauschens werden die Spektren in mehreren Stufen vorbehandelt. Zunächst werden sie durch Interpolation auf ein einheitliches Wellenzahlspektrum gebracht und „Spikes“ werden entfernt. Anschließend erfolgt die Eliminierung additiver und multiplikativer spektraler Effekte. Dabei werden verschiedene häufig verwendete Methoden der Normierung und Basislinienkorrektur nach ihren Risiken und Vorteilen für die vorliegende Aufgabenstellung beurteilt.

2.3.1.1 Methode der "Kleinsten-Quadrate"

Für einige Methoden der Datenvorbehandlung werden die Spektren durch mathematische Funktionen angenähert. Dabei wird der Zusammenhang zwischen einer unabhängigen Variable *x* (hier Wellenzahl) und einer abhängigen Variable *y* (hier gemessene Raman-Intensität) mathematisch beschrieben. In der Regel wird dafür die Methode der "Kleinsten-Quadrate" (engl. Least Squares Method) [43] verwendet, die auch „Ausgleichsrechnung“ oder „Fitting“ genannt wird. Die „Kleinste-Quadrate“ Schätzung ist das mathematische Standardverfahren in der Regressionsanalyse. Sie wird hier am Beispiel der univariaten linearen Regression vorgestellt. Die Methode findet aber ebenso bei der multivariaten linearen Regression (MLR) Anwendung, bei der ein linearer Zusammenhang zwischen mehreren unabhängigen Variablen und einer abhängigen Variablen hergestellt wird. Auch nichtlineare Zusammenhänge können modelliert werden, indem beispielsweise ein Polynom 2. oder höherer Ordnung als Funktion der „Kleinsten-Quadrate“ Schätzung zugrunde gelegt wird.

Ausgangspunkt der "Kleinste-Quadrate"-Schätzung ist also die Wahl einer geeigneten Funktion (z. B. Geradengleichung, Polynom definierten Grades), deren freie Parameter so optimiert werden, dass die Summe der quadrierten Abweichungen der Funktionswerte $\hat{y}$ von den experimentell ermittelten Messwerten *y* (engl. Residual Sum of Squares: RSS) minimiert wird. Für *m* gemessene Datenpunkte (hier Raman-Intensitäten an *m* verschiedenen Wellenzahlen eines Spektrums) erhält man folgendes Minimierungsproblem:

$$\mathrm{RSS} = \sum_{i=1}^{m} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{m} \varepsilon_i^2 \rightarrow \min! \qquad (2.2)$$

Dabei bezeichnet y_i den Messwert (hier Raman-Intensität) an der *i*-ten Ausprägung der unabhängigen Variable *x* (hier *i*-te Wellenzahl) und $\hat{y}_i$ den durch die Funktion *f*(*x*) geschätzten Wert an dieser Stelle ($\hat{y}_i = f(x_i)$). Die Abweichungen (Differenzen zwischen gemessenen Punkten und den Funktionswerten), die auch Residuen (engl. Residuals) genannt werden, sind in dem Fehlervektor $\boldsymbol{\varepsilon}$ enthalten.

Wie dieses Minimierungsproblem gelöst wird, hängt von der Art der Modellfunktion ab. Für den Fall einer univariaten linearen Regression ist die "Kleinste-Quadrate" Methode in Abbildung 2.4 veranschaulicht. Die zu optimierende Funktions-Gleichung lautet in diesem Fall:

$$\mathbf{y} = \beta_0 + \beta_1 \cdot \mathbf{x} + \boldsymbol{\varepsilon} \tag{2.3}$$

Dabei stellt der Regressionskoeffizient β_1 die Beziehung zwischen **x** und **y** her und entspricht der Steigung der Geraden. β_0 steht für den Achsenabschnitt und nimmt bei Zentrierung der Daten (Subtraktion des Mittelwertes des Vektors **x** von jedem Wert dieses Vektors) den Wert Null an.

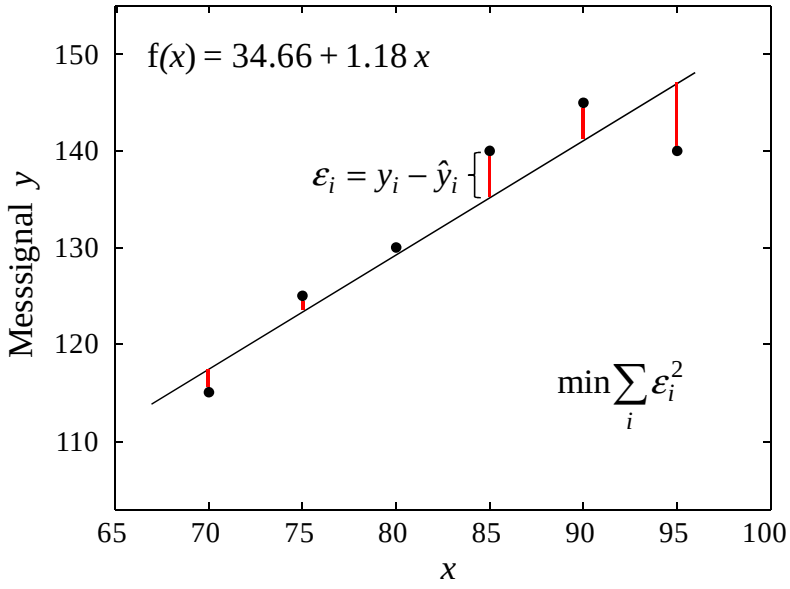


Abbildung 2.4: Lineare Regression durch Minimierung der Abweichungsquadrate

Durch Einsetzen von Ausdruck (2.3) in Gleichung (2.2) ergibt sich im Fall der linearen Regression folgendes Minimierungsproblem:

$$\mathrm{RSS} = \sum_{i=1}^{m} (y_i - (\beta_0 + \beta_1 \cdot x_i))^2 \rightarrow \min! \tag{2.4}$$

Durch partielle Ableitung von (2.4) nach β_0 und β_1, Nullsetzen und Auflösen nach β_0 und β_1 erhält man:

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{m} (x_i - \bar{x}) \cdot (y_i - \bar{y})}{\sum_{i=1}^{m} (x_i - \bar{x})^2} = \frac{\mathrm{Kovarianz}(\mathbf{x},\mathbf{y})}{\mathrm{Varianz}(\mathbf{x})} \tag{2.5}$$

$$\hat{\beta}_0 = \bar{y} - \beta_1 \cdot \bar{x} \tag{2.6}$$

In Matrixschreibweise lautet dies folgendermaßen:

$$\hat{\beta}_1 = (\mathbf{x}^{\mathrm{T}} \cdot \mathbf{x})^{-1} \cdot \mathbf{x}^{\mathrm{T}} \cdot \mathbf{y} \tag{2.7}$$

Für die Berechnung von $\hat{\beta}_1$ nach Gleichung (2.7) müssen die Daten allerdings vorher zentriert werden. Die Methode reagiert empfindlich auf Ausreißer in den Daten, da große Fehler ε_i durch das Quadrieren in der Zielfunktion (2.4) streng bestraft werden. Deshalb wurden zahlreiche Methoden der robusten Regression entwickelt [44,45]. Dabei wird üblicherweise die Summe der quadrierten Residuen durch eine Funktion ersetzt, die weniger Gewicht auf große Residuen legt.

2.3.1.2 Interpolation

Die dem Datensatz zugrundeliegenden spektroskopischen Messungen wurden im Zeitraum von mehreren Monaten und von verschiedenen Personen durchgeführt. Dies führt dazu, dass trotz morgendlicher Gerätejustierung die Spektren in ihren Anfangswellenlängen mehr oder weniger verschoben sind. Um dadurch entstehende systematische Fehler auszuschließen, werden die Raman-Spektren zunächst auf ein vorgegebenes Wellenzahlspektrum interpoliert. Unter Interpolation versteht man das Umrechnen diskreter Daten (hier Messwerte) in eine kontinuierliche Funktion (die sogenannte Interpolante oder Interpolierende), welche diese Daten abbildet. Auf diese Weise können Raman-Intensitäten für Wellenzahlwerte berechnet werden, die zwischen den eigentlichen Messpunkten liegen. Die von Isaac Newton begründete lineare Interpolation wird aufgrund ihrer Einfachheit in der Praxis am häufigsten verwendet. Dabei werden die Intensitätswerte zweier Datenpunkte x_0 und x_1 über eine Gerade miteinander verbunden. In dieser Arbeit wird auf das mittlere Wellenzahlspektrum aller Spektren linear interpoliert. Die obere und untere Wellenzahlgrenze der Spektren wird dabei so gewählt, dass keine Wellenlänge der Originalspektren außerhalb des zu interpolierenden Wellenzahlbereichs liegt. Das für die Interpolation verwendete Mittelwertsspektrum umfasst 941 Wellenzahlen im Bereich zwischen 3365 nm und 537 nm. Abbildung 2.5 zeigt Original-Raman-Spektren (Abbildung 2.5A) und Raman-Spektren nach Interpolation (Abbildung 2.5B). Interpolierte Spektren werden im Ergebnisteil der Arbeit mit INTERPOL gekennzeichnet.

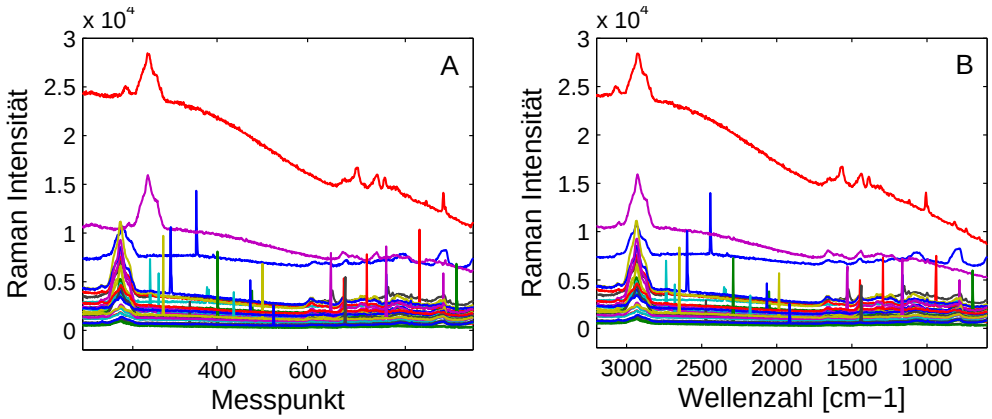


Abbildung 2.5: Ramanspektren vor (**A**) und nach (**B**) der Interpolation

In Abbildung 2.5A erkennt man, dass die Lage der CH-Streckschwinungsbanden (2700 bis 3000 cm^{-1}) für manche Spektren verschoben ist, was darauf hindeutet, dass die Messpunkte, an denen die Raman-Intensitäten der Spektren aufgenommen wurden, unterschiedlich waren. Nach der Interpolation (Abbildung 2.5B) befinden sich die Peaks bei allen Spektren an der gleichen Stelle.

2.3.1.3 „Spike"-Eliminierung

Nach der Interpolation der Spektren werden vorhandene „Spikes" entfernt. Der dafür verwendete Algorithmus basiert auf der Publikation von Philips und Harris [46]. In der hier angewendeten, modifizierten Fassung des Algorithmus werden „Spikes" bestehend aus bis zu 7 Datenpunkten erkannt und entfernt, während in der Originalpublikation nur „Spikes", die ein oder zwei aufeinanderfolgende Datenpunkte einnehmen, eliminiert werden können. Die „Spike"-Datenpunkte werden nach Erkennung durch eine einfache lineare Interpolation ersetzt. Bei Philips und Harris wird für die Interpolation ein Polynom 2. Ordnung verwendet. Sind in einem Spektrum keine „Spikes" enthalten, findet in der Originalpublikation eine Savitzky-Golay-Glättung [47] statt. In der hier verwendeten modifizierten Fassung dient der Algorithmus nach Savitzky und Golay nur dem Zweck, ungewöhnliche Punkte („Spikes") zu identifizieren. Diese werden anschließend entfernt, ohne dass das Spektrum geglättet wird.

Savitzky und Golay waren Pioniere in der Anwendung der lokalen „Kleinste-Quadrate" Schätzung (siehe Kapitel 2.3.1.1), mit dem Ziel mathematische Kurven zu glätten oder deren Ableitung zu bilden. Dabei wird zunächst eine Intervallgröße τ bestimmt. y_i sei ein beliebiger Messwert an dem Messpunkt x_i eines zu glättenden Spektrums. Durch die Spektrenwerte des Intervalls $\{x_{i-\tau}; x_{i+\tau}\}$ wird ein Polynom niedriger Ordnung mittels der „Kleinsten-Quadrate" Schätzung angenähert. Anschließend wird dem zentralen Intervallpunkt x_i der berechnete Polynomwert $\hat{y}_i$ zugeordnet. Dies wird für alle Datenpunkte des Spektrums wiederholt und man erhält die geglättete Kurve. Je größer das Intervall τ ist, desto stärker wird das Spektrum geglättet. Da nur der Zentralwert der geschätzten Polynome für die Glättung verwendet wird, schlugen Savitzky und Golay eine einfache und schnelle Methode vor, mit der durch eine gewichtete Linearkombination der Intervallpunkte der

zentrale Wert bestimmt werden kann, ohne dass das komplette Polynom berechnet werden muss.

Die Identifizierung der „Spikes“ mit Hilfe des Savitzky-Golay Algorithmus erfolgt durch die Berechnung der Residuen zwischen der nach Savitzky und Golay geglätteten Kurve und dem Originalspektrum. Sehr große Residuen werden als „Spikes“ erkannt und anschließend entfernt. Der modifizierte Algorithmus nach Phillips und Harris besteht aus folgenden Schritten:

Algorithmus 2.1: „Spike“-Eliminierung modifiziert nach Philips und Harris

1. Berechne geglättetes Raman-Spektrum mit einem Savitzky-Golay Filter (Polynom 2. Ordnung, Fensterbreite: 7 Punkte).
2. Berechne die standardisierten Residuen zwischen geglättetem Spektrum und Originalspektrum gemäß Gleichung (2.8). Das Originalspektrum wird dabei nicht verändert.
3. Definiere alle Datenpunkte deren standardisierte Residuen r_i / σ größer als ein benutzerdefinierter „Cut-Off“–Wert (hier 3.5) sind, als „Spikes“.
4. Ersetze „Spike“-Datenpunkte im Originalspektrum durch lineare Interpolation der an die „Spikes“ angrenzenden Datenpunkte.
5. Wiederhole Schritt 1-4 bis keine „Spikes“ mehr identifiziert werden.

Die standardisierten Residuen r_i / σ werden folgendermaßen berechnet:

$$r_i / \sigma = (y_i - \hat{y}_i) / \sigma \tag{2.8}$$

Dabei bezeichnet y_i den *i*-ten Originalpunkt und $\hat{y}_i$ denselben Datenpunkt nach Savitzky-Golay Filterung. Die Standardabweichung σ wird über die mediane absolute Abweichung vom Median (engl. Median Absolut Deviation: MAD) der Residuen geschätzt ($\hat{\sigma}$).

MAD stellt ein robustes Streuungsmaß dar und ist folgendermaßen definiert [48]:

$$\mathrm{MAD} = \mathrm{Median}(|\, r_i - \tilde{r}_{0.5}\,|)$$

Dabei bezeichnet $\tilde{r}_{0.5}$ den Median über alle Residuen r_i. Für normalverteilte Daten besteht folgender Zusammenhang zwischen der geschätzten Standardabweichung $\hat{\sigma}$ und MAD [48]:

$$\hat{\sigma} \approx \mathrm{MAD} \cdot 1.483 \qquad (2.9)$$

Für die „Spike“-Eliminierung werden vom Benutzer zwei Parameter definiert. Das sind zum einen die Fensterbreite des Savitzky-Golay-Filters und zum anderen der Grenzwert der standardisierten Residuen, ab dem ein „Spike“ als solcher identifiziert wird. Beide Parameter beeinflussen die Sensitivität der Methode. Je höher der „Cut-Off“-Wert der Residuen angesetzt wird, desto höher muss die „Spike“-Intensität sein, damit ein „Spike“ erkannt wird. Ist der „Cut-Off“-Wert zu niedrig, werden auch normale Peaks als „Spikes“ identifiziert, was zu einer Glättung der Peaks führt. Für die Fensterbreite des Savitzky-Golay-Filters gilt: Je größer das Fenster ist, desto stärker werden die Spektren, die mit den Originalspektren verglichen werden, im ersten Schritt des Algorithmus geglättet. Ist die Fensterbreite zu groß, werden auch spektrale Peaks als „Spikes“ definiert. Ist die Fensterbreite zu klein, werden „Spikes“ nicht erkannt. Es sei angemerkt, dass die zu wählende Fensterbreite für den Savitzky-Golay-Filter von der Auflösung der Spektren abhängt. So muss bei einer hohen Auflösung auch die Fensterbreite größer sein, um den gleichen Wellenzahlbereich abzudecken, wie bei einer niedrigeren Auflösung. Die Standardisierung der Residuen führt dazu, dass der benutzerdefinierte „Cut-Off“-Wert unabhängig von der Standardabweichung der Residuen ist und somit nicht für jedes Spektrum neu angepasst werden muss.
Abbildung 2.6 zeigt Spektren vor (Abbildung 2.6A) und nach (Abbildung 2.6B) der „Spike“-Eliminierung. Alle weiteren Vorbehandlungsmethoden werden auf interpolierten und „Spike“-eliminierten Spektren ausgeführt. „Spike“-eliminierte Spektren werden im Ergebnisteil der Arbeit mit SPIKEELIM gekennzeichnet.

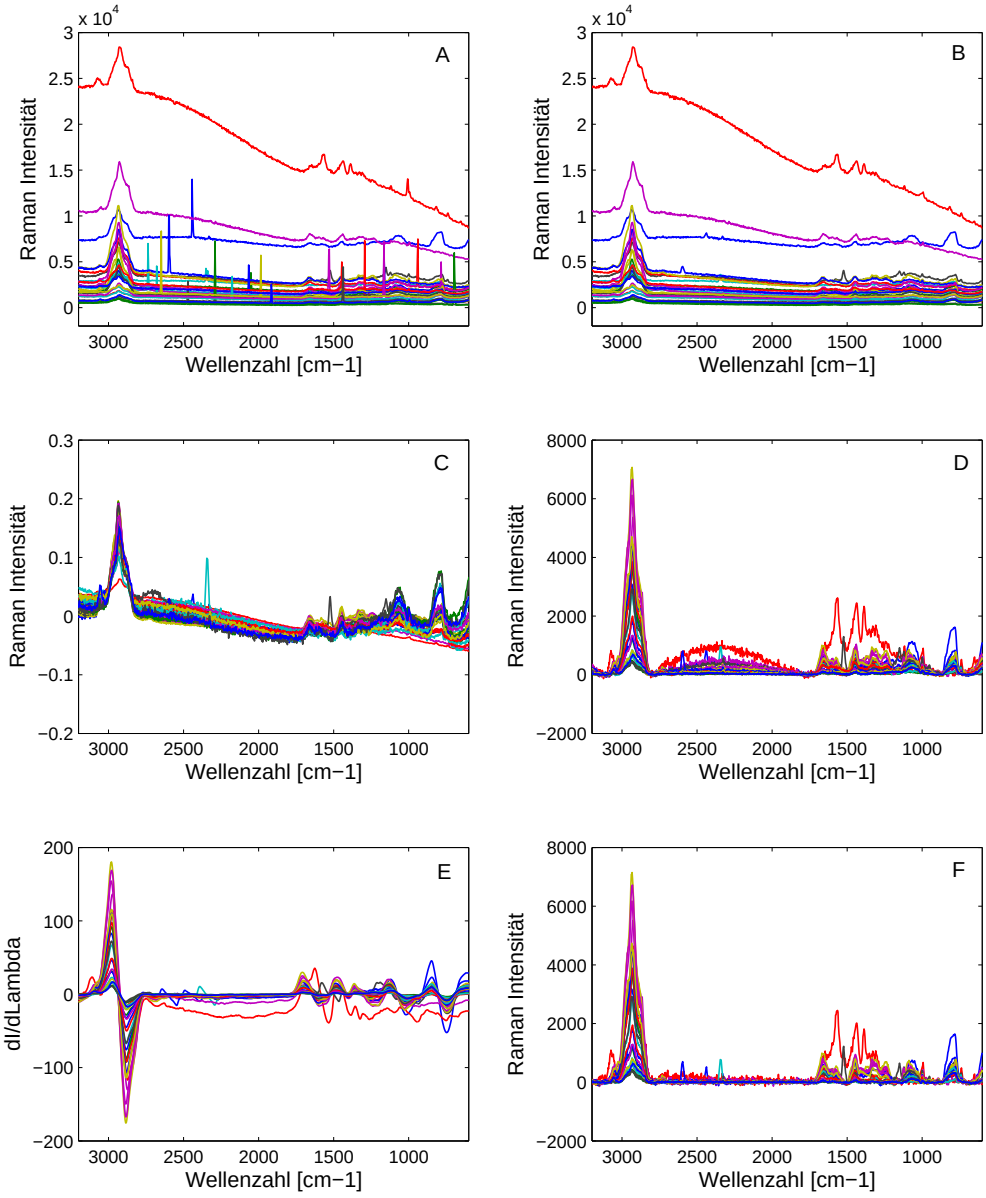


Abbildung 2.7: Methoden der Spektrenvorbehandlung. **A**: Rohspektren nach Interpolation. **B**: „Spike“-Eliminierung. **C**: Vektornormierung. **D-F:** Methoden der Basislinienkorrektur. **D**: Robuster Polynomfit (4. Ord.). **E**: 1. Ableitung. **F**: Whittaker Smoother.

2.3.1.4 Vektornormierung

Mit Hilfe der Vektornormierung (VEKNORM) können multiplikative Effekte in Spektren entfernt werden. Im Gegensatz zur Basislinienkorrektur (siehe Kapitel 2.3.1.5) wird dabei die Originalform der Spektren beibehalten. So werden zwei Spektren, bei denen die Banden das gleiche Verhältnis zueinander haben, aber für jedes Spektrum unterschiedliche maximale Intensitäten vorliegen, durch die Normierung identisch. Normierung kann auf verschiedene Arten erfolgen [43]. Häufig wird auf den Gesamtmittelwert eines Spektrums normiert. Bei Raman-Spektren biologischer Proben ist es außerdem möglich, auf den Mittelwert oder das Integral der CH-Streckschwingungspeaks bei 2700 bis 3000 cm^{-1} zu normieren, da deren Intensität proportional zum gesamten organischen Material ist. In dieser Arbeit wird jedes Spektrum auf die Vektorlänge Eins normiert. Dazu wird jeder Spektrenwert y_i durch die euklidische Norm des gesamten Spektrums dividiert (Gl. (2.10)).

$$y_i^{norm} = \frac{y_i}{\sqrt{\sum_{i=1}^{m} y_i^2}} \tag{2.10}$$

2.3.1.5 Basislinienkorrektur

a. Ableitungen: Eine der effektivsten Methoden, um unterschiedliche Basislinieneffekte zu kompensieren, ist die Bildung von Ableitungen [43]. Diese verstärken die spektrale Auflösung, da überlagernde Banden deutlicher erkennbar werden. Allerdings wird dadurch auch das Signal-Rausch-Verhältnis verschlechtert, was den Effekt der besseren Auflösung wieder kompensieren kann. Die Methode zur Berechnung der Ableitung hat einen entscheidenden Einfluss auf diese Effekte. Bei dem hier verwendeten Savitzky-Golay Algorithmus wird das Signal-Rausch-Verhältnis schlechter, je kleiner die Fensterbreite gewählt wird. Bei der Wahl einer zu großen Fensterbreite werden die Spektren dagegen stark verzerrt. Diese konkurrierenden Einflüsse müssen bei der Bestimmung der Parameter berücksichtigt werden. Durch die Bildung von Ableitungen verliert das Spektrum seine ursprüngliche Form, was nachfolgende Interpretationen erschwert. Trotzdem sind

Ableitungen wegen ihrer Einfachheit und Leistungsfähigkeit die beliebteste Methode, um Störeffekte aus Spektren zu entfernen. Zur Basislinienkorrektur wird in dieser Arbeit die 1. Ableitung verwendet. Die Berechnung erfolgt über einen Polynomfit nach Savitzky und Golay [47] (siehe Kapitel 2.3.1.3) (Polynom 2. Ordnung, Intervallgröße: 21 Punkte). In Analogie zur Glättung nach Savitzky und Golay wird bei der Berechnung der Ableitung für jeden Datenpunkt ein Polynom über $2\tau+1$ Datenpunkte entwickelt.

$$f(x) = \alpha + \beta \cdot x + \gamma \cdot x^2 + \delta \cdot x^3 + ... + \varepsilon \cdot x^a \tag{2.11}$$

Damit ist das Spektrum lokal mit einem Polynom des Grades *a* beschrieben, welches abgeleitet werden kann. Für die erste und zweite Ableitung ergibt sich:

$$\text{1. Ableitung}: \quad f'(x) = 0 + \beta + 2 \cdot \gamma \cdot x + 3 \cdot \delta \cdot x^2 + ... + a \cdot \varepsilon \cdot x^{a-1} \tag{2.12}$$

$$\text{2. Ableitung}: \quad f''(x) = 0 + 0 + 2 \cdot \gamma + 6 \cdot \delta \cdot x + ... + (a-1) \cdot a \cdot \varepsilon \cdot x^{a-2} \tag{2.13}$$

Aus den Gleichungen ist ersichtlich, dass durch die Bildung der 1. Ableitung eine konstante Basislinie α entfernt wird. Mit der zweiten Ableitung fallen dagegen lineare Effekte $\beta \cdot x$ weg usw. Im Ergebnisteil ist die Verwendung der 1. Ableitung mit 1.ABL gekennzeichnet.

b. Robuster Polynomfit: Neben der Bildung von Ableitungen können Basislinieneffekte auch über einen robusten Polynomfit korrigiert werden. Dazu wird das an *m* Punkten gemessene Spektrum **y**=(y_1 ,…,y_m) als Summe aus der eigentlichen chemischen Information und zusätzlich auftretenden Störungen gesehen, was durch folgende Gleichung beschrieben werden kann **y** = **b**+**e**. **b** beschreibt die Basislinie und **e** das tatsächliche Spektrum. Mit Hilfe eines Polynoms *a*-ter Ordnung wird die Basislinie **b** angenähert und anschließend von dem Spektrum **y** abgezogen. Zurück bleibt im Idealfall ein von Störsignalen befreites Spektrum **e**.

Die Schätzung der Basislinie über ein Polynom *a*-ter Ordnung kann mit der Gleichung $\mathbf{b}=\mathbf{H}\cdot\boldsymbol{\beta}$ beschrieben werden, wobei $\mathbf{H}$ (Vandermonde Matrix [49] der Wellenzahlen *h*) und $\boldsymbol{\beta}$ (Koeffizienten des Polynoms) folgendermaßen definiert sind:

$$\mathbf{H}(h_1, h_2, ..., h_m) = \begin{pmatrix} h_1^0 & \cdots & h_1^a \\ \vdots & & \vdots \\ h_m^0 & \cdots & h_m^a \end{pmatrix}, \boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \vdots \\ \beta_a \end{pmatrix}$$

Dabei beschreibt *m* die Anzahl der Messpunkte (Wellenzahlen) im ausgewählten Bereich. Somit entspricht h_1 dem ersten und h_m dem letzten Messpunkt des Spektrums. Der Exponent von *h* in der Wellenzahlen-Matrix $\mathbf{H}$ ergibt sich aus dem Kolonnenindex minus 1. Dieser nimmt höchstens den Wert *a* an, der dem Grad des Polynoms bei der Basislinienapproximation entspricht. Die Annäherung der Basislinie erfolgt durch die Minimierung der Funktion $\vartheta(\boldsymbol{\beta})$:

$$\vartheta(\boldsymbol{\beta}) = \sum_{i=1}^{m} \varphi(y_i - (\mathbf{H}\cdot\boldsymbol{\beta})_i) \rightarrow \min! \qquad (2.14)$$

Bei der "Kleinsten-Quadrate“ Schätzung ist die Kostenfunktion φ quadratisch. Dabei führen hohe Intensitätswerte (Peaks) zu sehr hohen Kosten, da sie quadriert in die Berechnung eingehen. Diese würden die Basislinienschätzung stark beeinträchtigen; d.h. im Bereich der Peaks würde auch die geschätzte Basislinie einen positiven Ausschlag aufweisen. Da spektrale Information und vor allem Peaks nicht oder nur in geringem Maß in die Schätzung eingehen sollen, wird eine robuste Form der "Kleinsten-Quadrate" Schätzung für die Annäherung der Basislinie verwendet. Es wird eine Kostenfunktion gewählt, die für kleine Residuen quadratisch ist, während bei großen positiven Residuen (Peaks) der Einfluss der gemessenen Intensitäten auf die Schätzung eliminiert wird. Übersteigt ein Residuum *r* einen definierten „Cut-Off“-Wert *s*, nimmt der Funktionswert $\varphi(r)$ den konstanten Wert s^2 an. Da dies bei negativen Abweichungen nicht gilt (bei negativen Abweichungen ist die Funktion quadratisch), wird die Kostenfunktion als asymmetrisch bezeichnet.

Die gewählte Kostenfunktion $\varphi(r)$, die in Abbildung 2.8 veranschaulicht ist, ist folgendermaßen definiert:

$$\forall r \in \Re,\ \varphi(r) = \begin{cases} r^2, & \text{wenn} \quad r < s \\ \\ \text{andernfalls} & s^2 \end{cases} \tag{2.15}$$

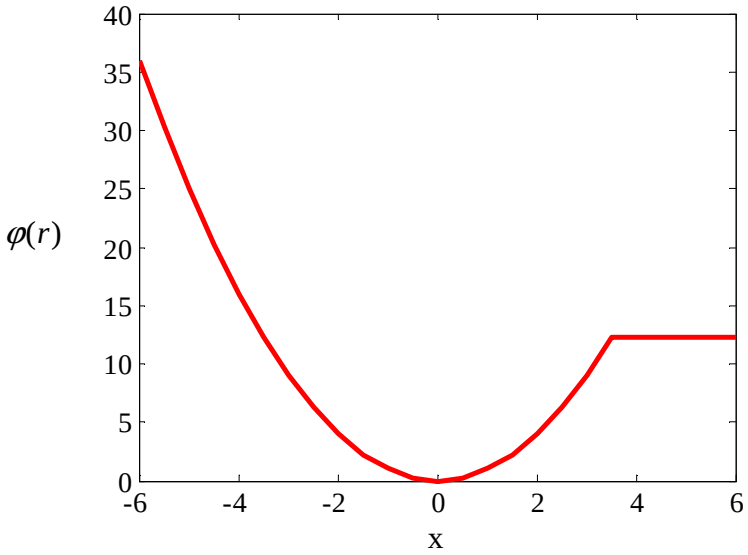


Abbildung 2.8: Kostenfunktion $\varphi(r)$ zur Schätzung der Basislinie des Raman-Spektrums. Bei negativen und kleinen Residuen r verhält sich die Kostenfunktion quadratisch. Bei positiven Residuen erhält man ab einem bestimmten „Cut-Off"-Wert s (hier 3.5) den Funktionswert s^2.

Die Minimierung von Gleichung (2.14) wird in dieser Arbeit iterativ durch eine halb-quadratische Minimierung nach Mazet und Kollegen gelöst [50].

Für die Basislinienkorrektur werden in dieser Arbeit Polynome verschiedener Ordnung (2., 4., 6., 8. Ordnung) mit verschiedenen „Cut-Off"-Werten berechnet. Nach visueller Inspektion der daraus resultierenden Spektren wird ein Polynom 4. Ordnung (POLY4) mit einem „Cut-Off"-Wert von 0.0001 für die weitere Analyse gewählt.

c. Asymmetrischer Whittaker-Smoother: Wie bei der Anwendung des Robusten Polynomfits wird auch beim Asymmetrischen Whittaker-Smoother die von Störsignalen behaftete Basislinie geschätzt und von dem Spektrum abgezogen. Für die Annäherung der Basislinie wird ein Glättungsalgorithmus nach Eilers verwendet, der auf einer „penalized“ "Kleinste-Quadrate" Schätzung basiert [51]. Die ursprüngliche Idee des Algorithmus stammt von Whittaker, der diesen schon im Jahr 1923 veröffentlichte [52]. 80 Jahre später wurde er von Eilers aufgegriffen und weiterentwickelt. „Penalized“ heißt aus dem Englischen ins Deutsche übersetzt „bestraft“. In diesem Kontext bedeutet der Ausdruck, dass neben den Abweichungsquadraten ein weiteres Kriterium in die zu minimierende Funktion eingeht. Die "Kleinste-Quadrate" Schätzung wird also von einem weiteren Kriterium, das hier der Grad der Glättung R ist, in einem vom Benutzer definierten Ausmaß „bestraft“. Somit stehen sich zwei konkurrierende Ziele gegenüber: (1) die Genauigkeit der Datenanpassung vertreten durch die Abweichungsquadrate und (2) die Glattheit der Kurve vertreten durch den Glättungsgrad R. Dieser Zusammenhang wird anhand eines Beispiels näher erläutert. Eine Kurve **b** wird auf ein Spektrum **y** gefittet. Dabei kann die Glattheit von **b** durch die Differenzen zwischen benachbarten Datenpunkten charakterisiert werden. Quadrieren und Aufsummieren dieser Differenzen ergibt ein effektives Maß für die Glattheit von **b** (Gl. (2.16) und (2.17)).

$$\Delta b_i = b_i - b_{i-1} \tag{2.16}$$

$$\mathrm{R} = \sum_i (\Delta b_i)^2 \tag{2.17}$$

Hier steht $\sum_i$ für die Summe des Terms über alle Messpunkte *i*.

Die Genauigkeit der Datenanpassung RSS wird durch die Abweichungsquadrate beschrieben:

$$\mathrm{RSS} = \sum_i (y_i - b_i)^2 \tag{2.18}$$

Eine Kombination der beiden konkurrierenden Terme ergibt:

$$Q = \mathrm{RSS} + \theta \cdot \mathrm{R} \tag{2.19}$$

$$Q = || \mathbf{y} - \mathbf{b} ||^2 + \theta \cdot || \mathbf{D} \cdot \mathbf{b} ||^2 \tag{2.20}$$

Dabei entspricht $|| ... ||^2$ der quadratischen Norm eines beliebigen Vektors. **D** beschreibt eine Matrix, so dass $\mathbf{D} \cdot \mathbf{b} = \Delta\mathbf{b}$. Beispielsweise wäre **D** für ein Spektrum, das aus 5 Datenpunkten besteht:

$$\mathbf{D} = \begin{pmatrix} -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 1 \end{pmatrix}$$

Für die Minimierung von Gl. (2.20) wird die partielle Ableitung gebildet (Gl. (2.21)). Diese wird gleich Null gesetzt und nach **y** aufgelöst. So erhält man Gl. (2.22), in der **I** die Einheitsmatrix bezeichnet.

$$\partial \mathrm{Q} / \partial \mathbf{b}^{\mathrm{T}} = -2 \cdot (\mathbf{y} - \mathbf{b}) + 2\theta \cdot \mathbf{D}^{\mathrm{T}} \cdot \mathbf{D} \cdot \mathbf{b} \tag{2.21}$$

$$(\mathbf{I} + \theta \cdot \mathbf{D}^{\mathrm{T}} \cdot \mathbf{D}) \cdot \mathbf{b} = \mathbf{y} \tag{2.22}$$

Aus Gründen der Verständlichkeit wurde zunächst der Glattheitsparameter $R = \sum_i (\Delta b_i)^2$ als Summe der Differenzen 1. Ordnung beschrieben. In der Praxis erweisen sich häufig Differenzen 2. oder 3. Ordnung als geeigneter. In dieser Arbeit werden Differenzen 2. Ordnung ($\Delta^2 b_i$) verwendet. Die zugehörigen Formeln lauten:

$$\Delta^2 b_i = \Delta(\Delta b_i) = (b_i - b_{i-1}) - (b_{i-1} - b_{i-2}) = b_i - 2b_{i-1} + b_{i-2} \qquad (2.23)$$

Da der Whittaker-Smoother hier nicht zur Glättung der Spektren verwendet wird, sondern der Schätzung der Basislinie dient, sollen spektrale Peaks, die chemische Information enthalten, nicht oder nur in einem geringen Ausmaß in die Schätzung eingehen. Aus diesem Grund kontrolliert ein Gewichtsvektor den Einfluss jedes Datenpunktes auf Gl. (2.22). Dabei bekommen negative Abweichungen (Rauschen) von der Basislinie ein höheres Gewicht als positive Abweichungen (Rauschen oder Peaks). Die asymmetrische Gewichtung führt zu der Bezeichnung „Asymmetrischer Whittaker-Smoother“. Positive Residuen, die einen „Cut-Off“-Wert überschreiten, werden durch einen konstanten Wert ersetzt, was den Einfluss des Peaks auf die Basislinienschätzung stark reduziert. Die vom Benutzer festzulegenden Parameter sind zum einen θ, welches das Ausmaß charakterisiert, mit dem das Kriterium „Glattheit“ R in die Gleichung eingeht. Daneben wird die positive/negative Gewichtung der Datenpunkte festgesetzt sowie der „Cut-Off“-Wert der Residuen, ab dem ein Spektrenwert nicht mehr in die Basislinienschätzung eingeht. Diese Parameter werden durch visuelle Inspektion der Spektren gewählt und bestehen aus folgenden Werten: θ=10^7, positive/negative Gewichte=0.25/0.75, „Cut-Off“-Wert=100. In Abbildung 2.7F sieht man, dass die so behandelten Spektren sehr gleichmäßige Basislinien aufweisen. Im Ergebnisteil wird der „Asymmetrische Whittaker-Smoother“ mit WHIT gekennzeichnet.

2.3.2 Dimensionsreduktion

In der Schwingungsspektroskopie sind die Daten im höchsten Maß multikollinear, d.h. die Variablen sind stark untereinander korreliert. Man kann dies leicht nachvollziehen, wenn man benachbarte Messpunkte eines Peaks betrachtet, deren Intensitäten sich ähnlich verhalten. Aus diesem Grund lässt sich ein Spektrum, das aus 941 Variablen bzw. Messpunkten besteht, tatsächlich durch wesentlich weniger als 941 Dimensionen beschreiben. Multikollinearität führt bei der Anwendung vieler Lernalgorithmen zu numerischen Problemen, was durch eine vorherige Dimensionsreduktion verhindert werden kann. Eine klassische Methode für diesen Zweck ist die Hauptkomponentenanalyse (engl. Principal Component Analysis: PCA) [53], die in Abbildung 2.9 schematisch dargestellt ist. Das Prinzip der PCA besteht in der Umrechnung der gemessenen Ausgangsdaten in neue sogenannte latente Variablen, die auch Faktoren, oder Hauptkomponenten (engl. Principal Components: PCs) genannt werden. Diese Faktoren sind eine Linearkombination der ursprünglichen Variablen; d.h. sie werden als Summe der unterschiedlich gewichteten Originalvariablen berechnet. Mathematisch wird bei der Hauptkomponentenanalyse eine Hauptachsentransformation durchgeführt. Dabei minimiert man die Korrelation zwischen den Variablen durch Überführung der Daten in einen Vektorraum mit neuer Basis. Das Koordinatensystem wird gedreht (siehe Abbildung 2.9). Die räumliche Anordnung der Datenpunkte zueinander verändert sich dabei nicht, die Datenpunkte erhalten allerdings neue Koordinaten - die sogenannten „Scores“. Durch die Transformation erhält man außerdem Informationen über die tatsächlich in den Daten vorhandene Dimensionalität.

Die Berechnung der PCA entspricht einem Eigenwertproblem. Gesucht werden die Eigenvektoren sowie die Eigenwerte der Varianz-Kovarianz-Matrix $\mathbf{\Sigma}$ (siehe Gl. (2.29)) der Datenmatrix **X**. **X** liegt in der Regel zentriert vor (vorheriges Abziehen der Mittelwerte der Kolonnenvektoren von den Einzelwerten der jeweiligen Vektoren). In den folgenden Kapiteln beschreibt **X** eine Matrix mit n Objekten (n Zeilen) und m Variablen (m Spalten). Die Objekte entsprechen hier den Raman-Spektren (n=3642) und die Variablen den Wellenzahlen (m=941).

Das Eigenwertproblem der PCA kann durch eine Singulärwertzerlegung (SVD) [54] gelöst werden. Dabei wird die zentrierte Datenmatrix **X** in zwei orthonormale Matrizen **U** und **V** sowie in eine Diagonalmatrix **S** zerlegt. Man erhält:

$$\mathbf{X} = \mathbf{U} \cdot \mathbf{S} \cdot \mathbf{V}^{\mathbf{T}} \tag{2.24}$$

Die Spalten von **V** entsprechen den Eigenvektoren der Varianz-Kovarianz-Matrix $\boldsymbol{\Sigma}$ von **X**. Diese Eigenvektoren werden auch als „Loadings“ bezeichnet. Die Diagonalwerte von **S** sind die Singulärwerte, die als Wurzeln der Eigenwerte der Varianz-Kovarianz-Matrix $\boldsymbol{\Sigma}$ definiert sind. Orthonormalität der Matrizen **V** und **U** bedeutet:

$$\mathbf{U}^{\mathbf{T}} \cdot \mathbf{U} = \mathbf{I}_n \quad \text{bzw.} \quad \mathbf{V}^{\mathbf{T}} \cdot \mathbf{V} = \mathbf{I}_m \tag{2.25}$$

$\mathbf{I}_n$ und $\mathbf{I}_m$ stellen Einheitsmatrizen der Dimension n x n bzw. m x m dar. Die Spalten von **V** bestehen also aus orthogonalen (d.h. unabhängigen bzw. unkorrelierten) Einheitsvektoren. Diese Einheitsvektoren zeigen in die Richtungen des neuen Koordinatensystems. Durch die Multiplikation der Matrix **X** mit **V** erfolgt die Rotation des ursprünglichen Koordinatensystems unter Bildung des neuen Koordinatensystem, dessen Hauptachsen die PCs darstellen. Die „Score“-Werte **T** beschreiben die Projektionen auf die jeweilige Hauptachse für jeden Datenpunkt. Unter Berücksichtigung von Gleichung (2.24) ergibt sich:

$$\mathbf{T} = \mathbf{X} \cdot \mathbf{V} = \mathbf{U} \cdot \mathbf{S} \cdot \mathbf{V}^{\mathrm{T}} \cdot \mathbf{V} = \mathbf{U} \cdot \mathbf{S} \tag{2.26}$$

Die Spalten in **T** entsprechen den Hauptkomponenten und sind dabei so angeordnet, dass die erste Hauptachse (erste Spalte) in die Richtung der größten Varianz der Daten zeigt. Die zweite Hauptachse steht senkrecht auf der ersten und es wird der zweithöchste Varianzanteil erklärt. Die weiteren Hauptkomponenten folgen mit abnehmender erklärter Varianz. Die Singulärwerte in **S** nehmen dementsprechend von links nach rechts ab; denn die erklärte Varianz einer Hauptkomponente ist gleich dem entsprechenden quadrierten Singulärwert in **S**. Daraus folgt auch die Anordnung der „Loadings“ in **V**.

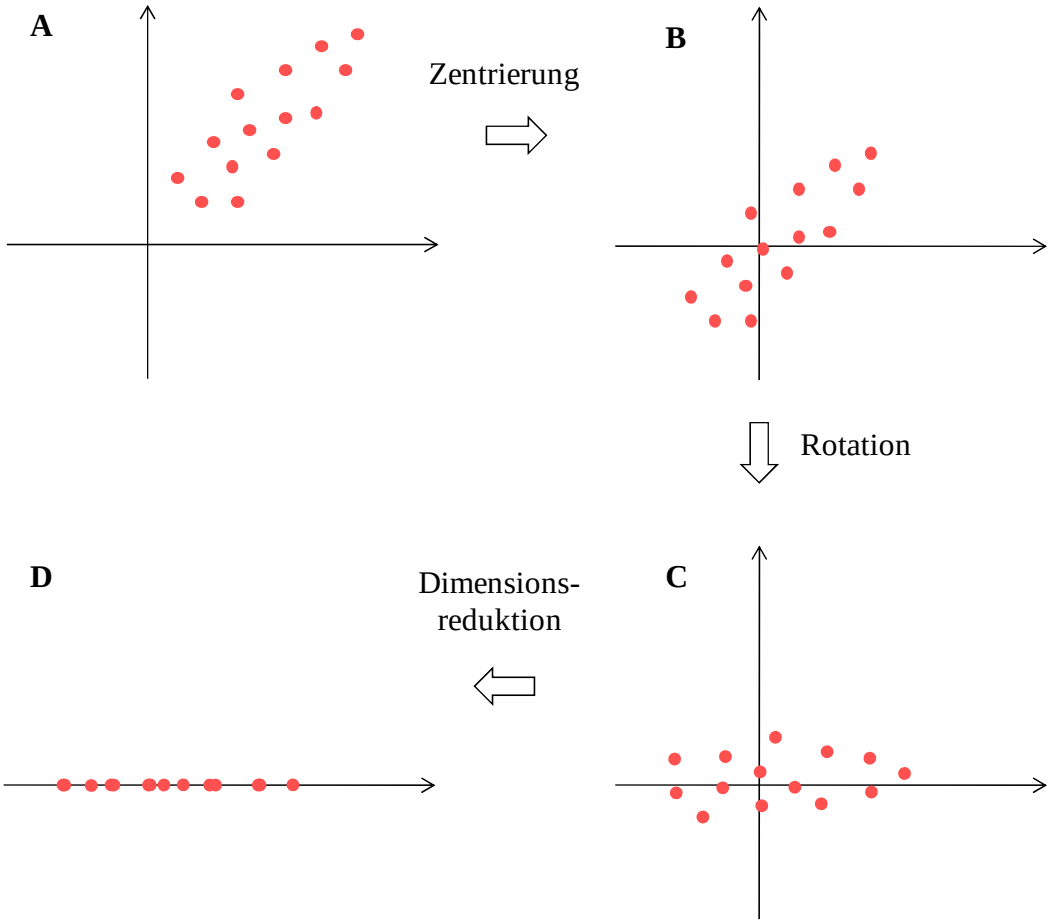


Abbildung 2.9: Geometrische Deutung der Hauptkomponentenanalyse. Von **A** nach **B** erfolgt die Zentrierung der Daten. **B** und **C** beschreiben die Drehung des Koordinatensystems, so dass die 1. Hauptachse (1. Hauptkomponente) in Richtung der höchsten Varianz der Daten zeigt. Reduziert man die Dimension auf eine Hauptkomponente, geht die Varianz der 2. Dimension verloren (**D**). Auf diese Weise kann man kleine Varianzen (Rauschen) eliminieren.

Normalerweise werden nur die ersten PCs für die folgende Analyse verwendet, da Hauptkomponenten mit sehr kleinen Singulärwerten in vielen Fällen Rauschen enthalten. Die Dimensionsreduktion hat also neben der Beseitigung von Multikollinearität auch den Zweck, Rauschen aus den Daten zu entfernen. Für die Klassifikation sollte man aber berücksichtigen, dass die ersten PCs nicht notwendigerweise die für die Klassifkation relevanteste Information enthalten. Dies lässt sich dadurch erklären, dass bei der PCA die Varianz akkumuliert wird,

während die Trennung der Klassen für die Berechnung der Hauptkomponenten keine Relevanz hat [55]. Diese Problematik besteht nicht bei der zur PCA verwandten Methode der "Partial Least Squares" (PLS) [56,57]. Bei der PLS wird die Klasseninformation für die Berechnung der latenten Variablen berücksichtigt. Normalerweise benötigt man deshalb weniger PLS-Komponenten für eine Klassifikation als PCs. PLS wird im Folgenden bei der Klassifikation mittels PLS-Diskriminanzanalyse (siehe Kapitel 2.3.3.4.1) verwendet.

Bei der Hauptkomponentenanalyse ist die Schätzung der geeigneten Anzahl an PCs ein entscheidender Schritt. Dabei besteht das Risiko des Informationsverlustes (Verwendung zu weniger PCs) oder der Einbeziehung von zufälligem Rauschen (Verwendung zu vieler PCs). Dies entspricht der in Kapitel 2.3.3.2 beschriebenen Problematik des "Under-" bzw. "Overfittings". Im ersten Fall ist die Anzahl der PCs zu gering, um die Datenstruktur ausreichend wiederzugeben. Im zweiten Fall wird die Datenstruktur zu genau abgebildet, so dass auch Rauschen in die Modellbildung mit eingeht. "Overfitting" ist wahrscheinlicher und kritischer zu bewerten als "Underfitting", weil dabei die Güte des erstellten Klassifikationsmodells überoptimistisch eingeschätzt wird (siehe Kapitel 2.3.3.2 und Abbildung 2.12). In dieser Arbeit wird die optimale Anzahl an PCs so gewählt, dass die Klassifikationsrate in der 50-fachen Kreuzvalidierung möglichst hoch wird. Mit dieser Strategie kann man "Underfitting" in der Regel ausschließen. Um das Ausmaß der Modellselektion möglichst gering zu halten, werden nur 12 verschiedene Stufen für die Anzahl der PCs getestet (5, 10, 15, ..., 60). Wenn der Gewinn an Vorhersagegenauigkeit durch Vergrößerung der Anzahl an PCs nur klein ist (<0.5%), wird die kleinere Anzahl an Hauptkomponenten gewählt. Dadurch wird die Gefahr des "Overfittings" verringert, die bei einer größer werdenden Anzahl an PCs zunimmt. Bei Verwendung unterschiedlicher Vorbehandlungsmethoden für einen Klassifikator wird die Anzahl an PCs verwendet, die bei der Kombination der verschiedenen Vorbehandlungsmethoden mit dem Klassifikator am häufigsten gewählt wird.

Um nach Dimensionsreduktion und Klassifikation die Güte des Modells realistisch einschätzen zu können, ist es notwendig ein externes Testset für die Vorhersage zu verwenden, das unabhängig von der Parameteroptimierung (Wahl der Anzahl an PCs) ist. Aus diesem Grund wird ein doppeltes Validierungsschema verwendet, das in Kapitel 2.3.4.1.3 beschrieben ist.

2.3.3 Klassifikation

Die Klassifikation stellt die wichtigste Stufe in der Auswertung der Raman-Spektren dar. Hier werden die Spektren gemäß ihrer bakteriellen Stammzugehörigkeit differenziert. Aus diesem Grund liegt in dieser Arbeit ein besonderes Augenmerk auf den verwendeten Klassifikationsmethoden. Sie werden im Folgenden in verschiedener Hinsicht auf ihre Nützlichkeit für die Differenzierung von Bakterien mittels Mikro-Raman-Spektroskopie untersucht. Zum besseren Verständnis der bei einer Klassifikation ablaufenden Vorgänge wird zunächst der Begriff Klassifikation definiert sowie einige grundlegenden Prinzipien der Klassifikation erklärt. Im Anschluss daran werden die in dieser Arbeit verwendeten Klassifikationsmethoden vorgestellt.

Unter Klassifikation versteht man die systematische Zuordnung von Objekten zu Klassen oder Kategorien. Die Klassenzugehörigkeit der Objekte ist dabei im Vorfeld bekannt und geht in die Berechnung ein. Deswegen spricht man auch von überwachtem Lernen. Im Gegensatz dazu wird eine Gruppenbildung, die ohne vorherige Klasseninformation nur aufgrund der Datenstruktur selbst erfolgt, als unüberwachtes Lernen bezeichnet. Ein Beispiel für unüberwachtes Lernen ist die Clusteranalyse, die in Kapitel 2.3.5 beschrieben wird. Bei der Klassifikation unterscheidet man die Trainingsphase und die Testphase. In der Trainingsphase konstruiert der Lernalgorithmus basierend auf den gezeigten Trainingsdaten ein Modell. In der Testphase liefert dieses Modell für ein neues Objekt $\mathbf{x}_{neu}$ eine Ausgabe j, die der vorhergesagten Klassenzugehörigkeit entspricht. j ist ein Wert aus der Menge aller möglichen Klassen.

2.3.3.1 Distanz- und Ähnlichkeitsmaße

Sowohl beim überwachten als auch beim unüberwachten Lernen wird angenommen, dass sich Objekte derselben Klasse ähnlich und Objekte unterschiedlicher Klassen unähnlich sind. Klassifikationsalgorithmen basieren also auf der Berechnung der Ähnlichkeit der Objekte zueinander. Als Input für die Ähnlichkeitsberechnung dienen die Merkmalsvektoren der Objekte, die in dieser Arbeit den gemessenen Raman-Intensitäten der Spektren entsprechen. Die zwei gebräuchlichsten Abstandsmaße -die euklidische Distanz und die Mahalanobis-Distanz- werden im Folgenden kurz vorgestellt.

2.3.3.1.1 Euklidische Distanz

Die euklidische Distanz (ED) beschreibt den direkten Abstand zwischen zwei Punkten bzw. Vektoren im zwei- oder mehrdimensionalen Raum. Sie ist das am häufigsten verwendete Distanzmaß und wird für zwei Vektoren **x** und **y** im *m*-dimensionalen Datenraum (hier *m* Messpunkte bzw. Wellenzahlen) folgendermaßen berechnet:

$$\mathrm{ED}(\mathbf{x},\mathbf{y}) = \sqrt{\sum_{i=1}^{m}(x_i - y_i)^2} = ||\,\mathbf{x}-\mathbf{y}\,|| \tag{2.27}$$

2.3.3.1.2 Mahalanobis-Distanz

Im Gegensatz zur euklidischen Distanz werden bei der Berechnung der Mahalanobis-Distanz (MD) [58] die Korrelationen in den Daten berücksichtigt. Wie in Kapitel 2.3.2 (Dimensionreduktion) beschrieben, sind in Raman-Spektren die Intensitäten benachbarter Wellenzahlen stark korreliert, was zu ellipsoiden Verteilungen im Datenraum führt. In einem solchen Fall ist die Mahalanobis-Distanz ein zuverlässigeres Abstandsmaß als die euklidische Distanz (siehe Abbildung 2.10). Die MD zwischen zwei Vektoren $\mathbf{x}_1$ und $\mathbf{x}_2$ im *m*-dimensionalen Datenraum ist definiert als:

$$\mathrm{MD}(\mathbf{x_1},\mathbf{x_2}) = \sqrt{(\mathbf{x_1}-\mathbf{x_2})^{\mathrm{T}}\cdot\boldsymbol{\Sigma}^{-1}\cdot(\mathbf{x_1}-\mathbf{x_2})} \tag{2.28}$$

Dabei bezeichnet $\boldsymbol{\Sigma}$ die Varianz-Kovarianz-Matrix, die aus den Daten geschätzt wird. Das "Hütchen" ($\hat{\boldsymbol{\Sigma}}$) kennzeichnet, dass es sich um einen geschätzten Parameter handelt. Für eine spaltenzentrierte Matrix **X**, die *n* Objekte (*n* Zeilen) und *m* Variablen (*m* Spalten) enthält, ist die Varianz-Kovarianz-Matrix definiert als

$$\hat{\boldsymbol{\Sigma}} = \frac{1}{n-1}\cdot\mathbf{X}^{\mathbf{T}}\cdot\mathbf{X} \tag{2.29}$$

Der Effekt, der durch das Einbeziehen der Varianz-Kovarianz-Matrix in die Distanzberechnung entsteht, ist in Abbildung 2.10 veranschaulicht.

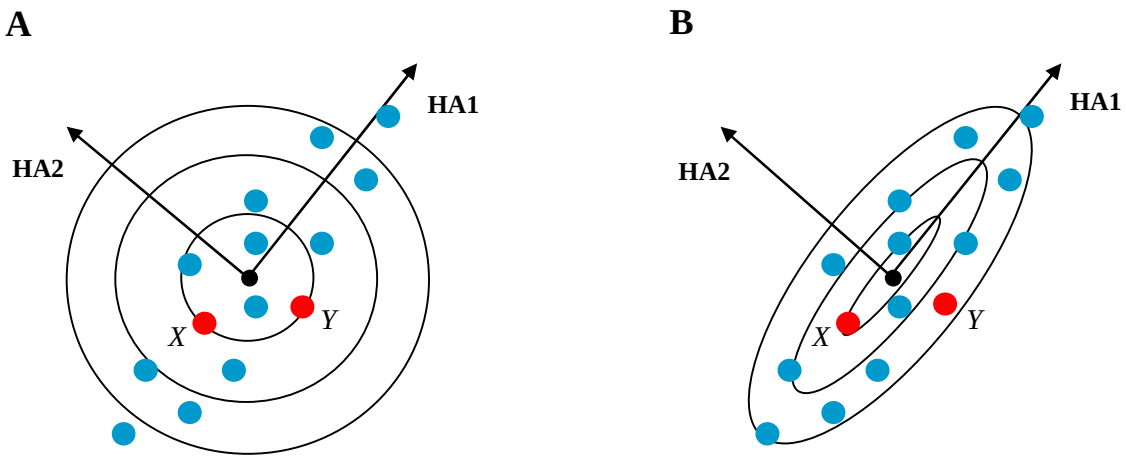


Abbildung 2.10: Vergleich von Euklidischer Distanz (**A**) und Mahalanobis-Distanz (**B**). HA1 (Hauptachse 1) und HA2 (Hauptachse 2) entsprechen den zwei Hauptachsen der Ellipse im Datenraum.

X und *Y* stellen in der Abbildung zwei Punkte einer Klasse dar. Die Datenpunkte sind nicht sphärisch sondern ellipsoid um den Klassen-Schwerpunkt bzw. Zentroiden verteilt, was auf korrelierende Variablen hinweist. Wie man in Abbildung 2.10 sieht, ist die Ausdehnung der Punktwolke in Richtung der 2. Hauptachse (HA2) (Richtung der geringeren Korrelation) weniger wahrscheinlich als entlang der 1. Hauptachse (HA1) (Richtung der höchsten Korrelation). Punkt *Y* liegt bezüglich des Zentroiden in die Richtung der 2. Hauptachse der Ellipse, während Punkt *X* in die Richtung der 1. Hauptachse liegt. Bei Berechnung der euklidischen Distanz der Punkte *X* und *Y* zum Klassen-Schwerpunkt erhält man den gleichen Abstand für beide Punkte (1 Einheit). Bei Berechnung der Mahalanobis-Distanzen hingegen weist Punkt *Y* einen wesentlich höheren Abstand vom Zentroiden (2.5 Einheiten) auf als *X* (1 Einheit), was aufgrund der Korrelation sinnvoll ist.

In dieser Arbeit spielen die euklidische Distanz sowie die Mahalanobis-Distanz bei verschiedenen Algorithmen des überwachten und des unüberwachten Lernens eine Rolle. So

ist die euklidische Distanz das Abstandsmaß, auf dem hier der *k*NN-Algorithmus (siehe Kapitel 2.3.3.4.5) basiert. Die ED ist außerdem die Grundlage bei der Erstellung der Kohonen-Karten (siehe Kapitel 2.3.5.2) sowie beim *k*-Means-Algorithmus (siehe Algorithmus 2.3). Auf der Mahalanobis-Distanz basieren die Klassifikationsalgorithmen LDA, QDA und MDA (siehe 2.3.3.4), wobei bei der MDA zunächst eine Initialisierung der Modellparameter mit Hilfe des *k*-Means Algorithmus erfolgt, bei dem die ED als Abstandsmaß verwendet wird (siehe Kapitel 2.3.3.4.4).

2.3.3.2 Datenstruktur und Klassifikation

An zahlreichen Stellen in dieser Arbeit wird über die Verwendung von linearen und nichtlinearen Klassifikationsmethoden diskutiert. Diese Beschreibung bezieht sich auf die Fähigkeit der Klassifikationsmethoden lineare bzw. nichtlineare Entscheidungsgrenzen im Datenraum bilden zu können. Einige grundsätzliche Möglichkeiten zur Bildung von Entscheidungsgrenzen in der Klassifikation werden hier vorgestellt. Die Zweckmäßigkeit einer Klassifikationsmethode richtet sich in erster Linie nach der vorhandenen Datenstruktur. In [42] sind verschiedene Fälle von relevanten Datenstrukturen für eine Klassifikation vereinfacht dargestellt. Diese sind in Abbildung 2.11 wiedergegeben.

- Im einfachsten Fall (Abbildung 2.11A) ist eine lineare Trennung möglich. Eine korrekte Klassenzuordnung erhält man durch die Berechnung der euklidischen Distanz der Objekte zum Klassenschwerpunkt.
- Auch in Abbildung 2.11B führt eine lineare Entscheidungsgrenze zum Erfolg. Aufgrund der Korrelation der Variablen ist allerdings eine vollständig korrekte Klassenzuordnung durch die euklidische Distanz nicht zu erreichen. Dagegen führt die Berechnung der Mahalanobis-Distanzen zu den Klassenschwerpunkten zu dem gewünschten Ergebnis (siehe Kapitel 2.3.3.4.2 Lineare Diskriminanzanalyse).
- Abbildung 2.11C zeigt ein Klassifikationsproblem, das nicht mit einer linearen Entscheidungsgrenze zu lösen ist. Hier ist eine gekrümmte Diskriminanz-Funktion notwendig (z. B. Quadratische Diskriminanzanalyse, siehe Kapitel 2.3.3.4.3).

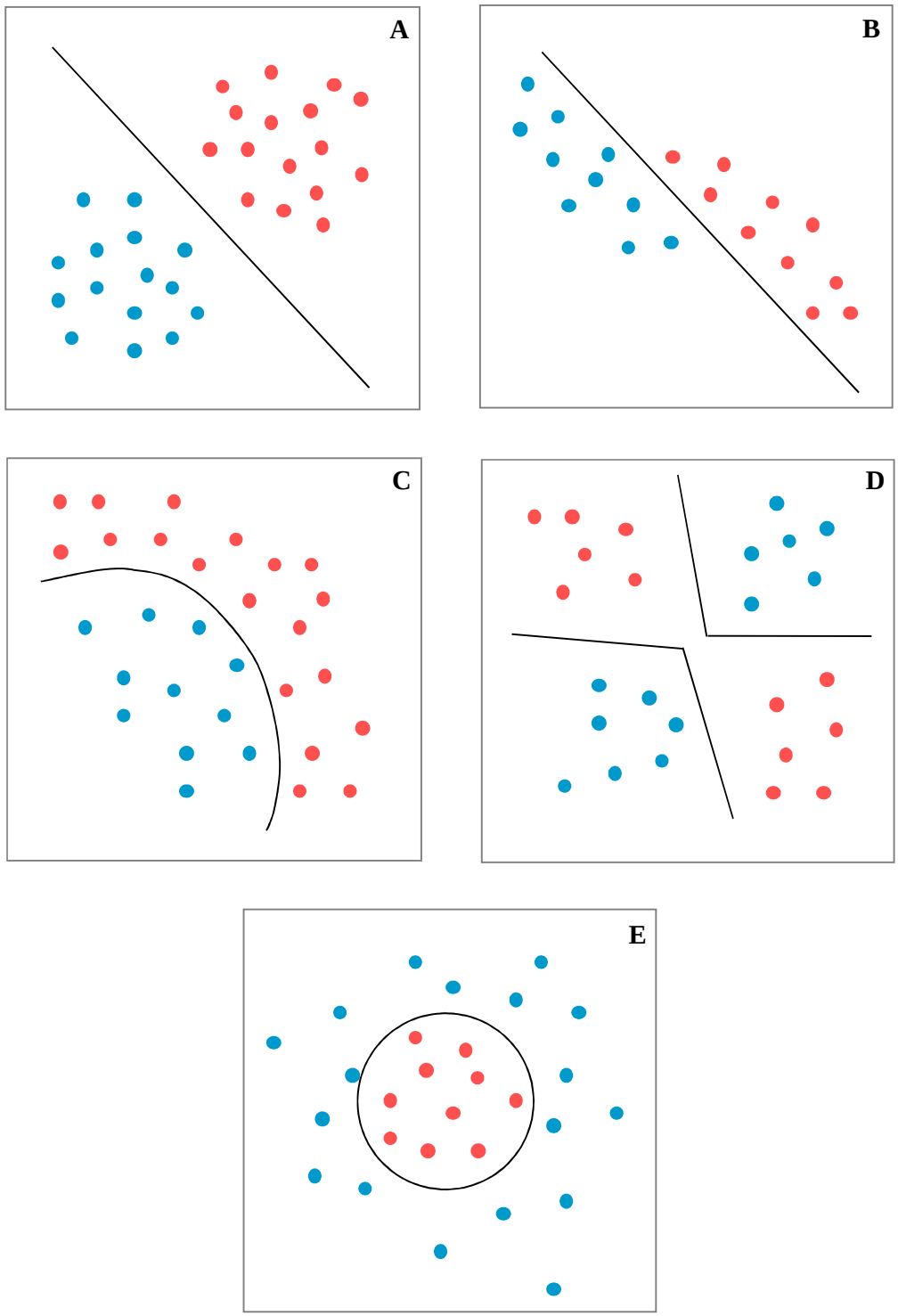


Abbildung 2.11A-E: Unterschiedliche Datenstrukturen in der Klassifikation (Zwei-Klassen-Fall) (nach [42]).

- In Abbildung 2.11D bestehen die Klassen aus mehreren Grüppchen bzw. Unterklassen. Eine geeignete Klassifikationsmethode muss sowohl nichtlineare Entscheidungsgrenzen bilden als auch einzelne streuende Gruppen zu einer Klasse zuordnen können (z. B. "Gaussian Mixture" Diskriminanzanalyse, *k*-nächste Nachbarn Klassifizierer, „Support Vector Machines“, siehe Kapitel 2.3.3.4).
- In Abbildung 2.11E ist eine Klasse von Objekten von einer anderen Klasse umgeben. Auch für dieses Klassifikationsproblem sind nichtlineare Entscheidungsgrenzen notwendig (z.B. "Gaussian Mixture" Diskriminanzanalyse, *k*-nächste Nachbarn Klassifizierer, Support Vector Machines, siehe Kapitel 2.3.3.4).

Intuitiv erscheint uns die Klassifikationsmethode am geeignetsten, die in der Lage ist, sich einer gegebenen Datenstruktur möglichst genau anzupassen. Man würde also erwarten, dass mit steigender Flexibilität die Vorhersagekraft einer Methode zunimmt. Eine intensive Modelloptimierung birgt jedoch die Gefahr, dass das gelernte Modell nur die Merkmale der Beispiele aus dem Training beschreibt. Für ungesehene Testobjekte, die den Trainingsdaten nicht exakt entsprechen, ist das Modell dagegen häufig unbrauchbar [59]. In diesem Fall spricht man von „Overfitting“ (Überanpassung). Dabei geht die Generalisierungsfähigkeit des Modells verloren. Dagegen wird es als „Underfitting“ (Unteranpassung) bezeichnet, wenn die Datenanpassung zu grob ist. In diesem Fall ist das Modell zu wenig flexibel für die gegebene Datenstruktur. Abbildung 2.12 zeigt Beispiele für „Underfitting“ und „Overfitting“ in der Klassifikation. In Abbildung 2.12A sind die Daten mit einem linearen Modell beschrieben, das relativ unflexibel ist. Das Modell passt sich den Daten grob an („Underfitting“). Die Vorhersage von ungesehenen Testobjekten liefert normalerweise den Vorhersagewert, der auf Basis der Trainingsdaten geschätzt wurde. Mit einer etwas genaueren Anpassung an die Daten könnte die Vorhersagegenauigkeit allerdings noch verbessert werden. Im Gegensatz dazu zeigt Abbildung 2.12B ein sehr flexibles Klassifikationsmodell, das sich den Daten so genau anpasst, dass für die Trainingsdaten keine einzige Fehlklassifikation vorkommt. Bei der Vorhersage des neuen Testobjektes ?, das wahrscheinlich zur blauen Klasse gehört, hat das Modell allerdings Probleme. Es ist nicht generalisierungsfähig („Overfitting“).

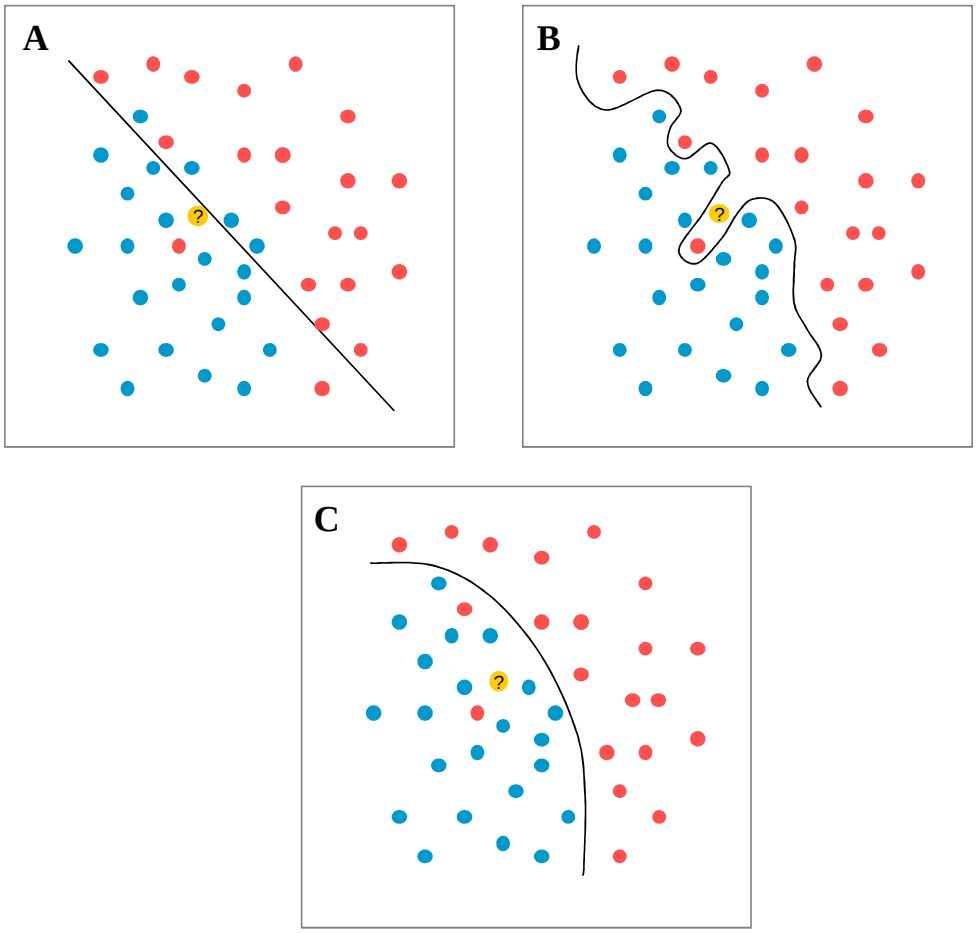


Abbildung 2.12: „Unterfitting“ (**A**) und „Overfitting“ (**B**) am Beispiel der Klassifikation. Das optimale Klassifikationsmodell ist in **C** gezeigt. Hier ist eine gute Balance zwischen Datenanpassung und Robustheit erreicht. ? beschreibt ein neues Testobjekt, das vorhergesagt werden soll.

Die optimale Lösung zeigt Abbildung 2.12C. Das Modell ist flexibler als das lineare Modell allerdings nicht so komplex, dass es sich exakt den Trainingsdaten anpasst. Die Generalisierbarkeit bleibt erhalten und das Testobjekt wird richtig zugeordnet.

Es ist also das Ziel der Klassifikation, den Modellfehler so zu minimieren, dass eine Balance zwischen Generalisierbarkeit und Vorhersagegenauigkeit auf den Trainingsdaten erreicht

wird. Alternativ spricht man dabei auch von einem „Bias-Varianz-Dilemma“. Mit „Bias“ wird ein systematischer Fehler bzw. die Verzerrung eines Modells bezeichnet. Im folgenden Beispiel kommt der „Bias“ durch eine schlechte Datenanpassung („Underfitting“) zustande. Abbildung 2.13 zeigt das typische Verhalten des Vorhersagefehlers von Trainingsdaten und Testdaten, wenn die Modellkomplexität zunimmt.

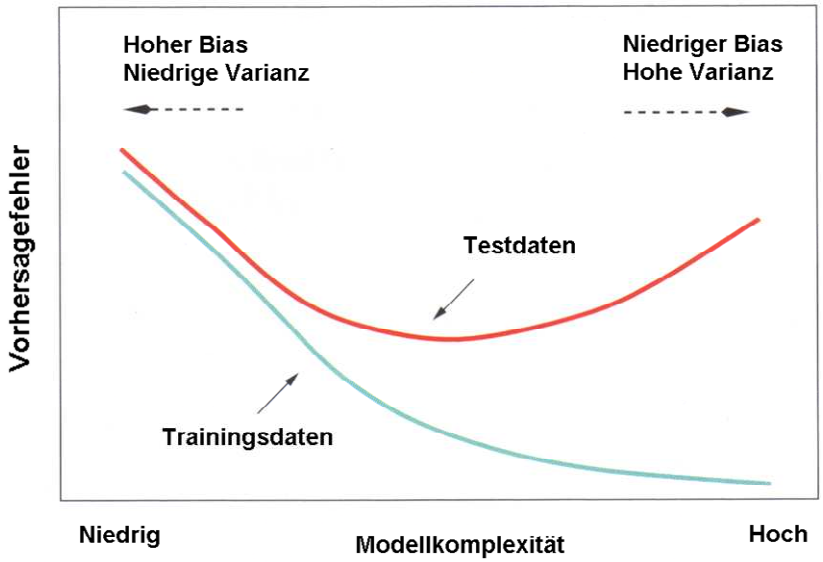


Abbildung 2.13: Veranschaulichung des „Bias-Varianz-Dilemma“ (verändert nach [60]). Mit zunehmender Modellkomplexität sinkt die Fehlerrate bzgl. der Daten, an die das Modell angepasst wurde. Für externe Testdaten ist der Fehler zum einen höher als für die Trainingsdaten, zum anderen nimmt der Optimismus, d.h. die Differenz zwischen den Fehlerraten auf Trainings- und Testdaten, mit steigender Modellkomplexität zu.

Bei zu niedriger Modellkomplexität beobachtet man „Underfitting“. Deshalb sind die Vorhersagewerte sowohl für die Trainingsdaten als auch für die Testdaten schlecht. Der „Bias“ ist hoch (schlechter Fit der Datenpunkte). Andererseits ist das Modell sehr robust; d.h. bei Veränderung der Zusammensetzung des Trainings-Datensatzes variiert das Modell nur geringfügig (kleine Varianz). Bei zunehmender Komplexität passt sich der Klassifikator den Daten immer stärker an. Der „Bias“ nimmt ab (guter Fit der Datenpunkte), was zunächst zu einer Abnahme des Vorhersagefehlers führt. Ab einem bestimmten Punkt nimmt jedoch der

Vorhersagefehler der Testdaten wieder zu, während der Fehler der Trainingsdaten weiter abnimmt. Dies entspricht einem klassischen „Overfitting“-Szenario. Das Modell verliert an Generalisierungsfähigkeit und Robustheit und es variiert stark für unterschiedlich zusammengesetzte Trainingsdaten (hohe Varianz). Der geringe Vorhersagefehler der Trainingsdaten vermittelt ein überoptimistisches Bild; denn die Testdatenvorhersage zeigt wesentlich schlechtere Werte als die Trainingsdatenvorhersage. In der Praxis ist das "Overfitting" wesentlich kritischer, als das "Underfitting", da es zu einer Fehleinschätzung des Modells kommt, was beim "Underfitting" nicht der Fall ist. Faktoren, die in dieser Arbeit zu einem „Overfitting“ führen können, sind zum einen die Wahl des Klassifikationsalgorithmus und zum anderen die Auswahl der „Tuning“-Parameter. Unter „Tuning“-Parameter werden hier alle Parameter verstanden, die vom Benutzer im Vorfeld der Klassifikation definiert werden (Anzahl der latenten Variablen für PLS-DA, Anzahl der PCs für LDA, MDA, QDA und *k*NN, *C* und γ für SVMs, Anzahl der Subzentren für MDA und *k* für *k*NN). Je flexibler der verwendete Klassifikationsalgorithmus ist, desto stärker ist die Gefahr des „Overfittings“. Die Flexibilität der Algorithmen wird von den „Tuning“-Parametern reguliert, weswegen deren Auswahl hier im Zusammenhang mit „Overfitting“ genauer betrachtet wird (siehe Kapitel 3.4.4.1). Die beiden Phänomene „Underfitting“ und „Overfitting“ lassen sich nur durch eine geeignete Validierung vermeiden. Dabei ist die Bestimmung des Vorhersagefehlers für ein externes Testset unverzichtbar. Auf geeignete Methoden der Validierung wird in Kapitel 2.3.4.1 näher eingegangen.

Neben „Overfitting“ neigen komplexe Modelle in vielen Fällen zu einer schlechten Interpretierbarkeit. Da flexible Entscheidungsgrenzen mathematisch komplexere Funktionen erfordern als lineare Klassifikationen, sind diese oft relativ undurchsichtig, was in manchen Fällen unangenehme Folgen nach sich ziehen kann. Ein bekanntes Beispiel für diese Problematik sind Neuronale Netze. So wurde in den 80er Jahren versucht, ein neuronales Netz zu entwickeln, das Panzer auf Bildern erkennen kann, die sich hinter Bäumen versteckt halten. Das Projekt war zunächst sehr erfolgreich. Es konnte sehr gut zwischen Bildern mit und ohne Panzer unterschieden werden. Später erkannte man jedoch den Grund für die hervorragenden Leistungen des neuronalen Netzes: Da alle Bilder mit verstecktem Panzer an einem bewölkten Tag entstanden waren, Bilder ohne Panzer jedoch an einem sonnigen Tag aufgenommen wurden, erkannte das Netz lediglich, ob der dargestellte Himmel bewölkt oder

sonnig war [61]. Auch „Support Vector Machines“ (SVMs) sind mathematisch komplex und deswegen schwer interpretierbar. Derartige Modelle prägten die Bezeichnung „Blackbox-Models“. Diese stehen einfachen und leicht interpretierbaren Klassifikationsmethoden wie z. B. der linearen Diskriminanzanalyse (LDA) (siehe Kapitel 2.3.3.4.2) gegenüber. So lässt die LDA aufgrund von Variablengewichtungen Rückschlüsse über die für die Klassifikation relevanten Variablen zu. Da für einige Klassifikationsprobleme eine lineare Entscheidungsgrenze nicht ausreichend ist, müssen häufig flexiblere Methoden herangezogen werden. Dabei ist es wünschenswert, eine Balance zwischen Flexibilität, Robustheit und Interpretierbarkeit zu finden. Dieses Ziel soll in der hier vorgestellten Arbeit unter Berücksichtigung verschiedener multivariater Techniken für die Klassifikation von Reinraumbakterien im „Online-Monitoring“ erreicht werden.

2.3.3.3 Klassifikationsrisiko

Da eine perfekte Klassifikationsrate häufig unmöglich zu erreichen ist, sind Maßzahlen, die das Risiko einer Fehlklassifikation beschreiben hilfreich für eine Entscheidungsfindung. Daher geben moderne Klassifikationsmethoden zusätzlich zu jeder Entscheidung einen Wert aus, der die Vertrauenswürdigkeit (Konfidenz) der getroffenen Entscheidung angibt. Dieser Wert wird als *a posteriori* Wahrscheinlichkeit bezeichnet. In Szenarien, in denen eine falsche Zuordnung zu einer Klasse schwerwiegendere Nachteile mit sich bringt als eine falsche Zuordnung zu einer anderen Klasse, ist es sinnvoll kostenoptimale Entscheidungsregeln zu verwenden, bei denen verschiedene Arten der Fehlklassifikation unterschiedlich gewichtet werden. Darüber hinaus kann es sinnvoll sein, einige Objekte als „nicht klassifizierbar“ zu deklarieren. Derartige Überlegungen sind Gegenstand der Entscheidungstheorie, einem Zweig der angewandten Wahrscheinlichkeitstheorie, die Konsequenzen von Entscheidungen evaluiert. Ein grundlegender statistischer Ansatz für die Klassifikation von Objekten ist die BAYES-Entscheidungstheorie [62]. Nach Bayes wird angenommen, dass das Entscheidungsproblem vollständig durch Wahrscheinlichkeiten beschrieben werden kann und dass alle relevanten Wahrscheinlichkeitswerte bekannt sind. So beschreibt die *a priori* Wahrscheinlichkeit $Pr(C_j)$ die grundsätzliche Wahrscheinlichkeit, dass ein Objekt der Klasse j angehört, ohne weitere Information über die Eigenschaften des zu klassifizierenden

Objektes zu berücksichtigen. $Pr(C_j)$ ist definiert als der zur Klasse j gehörende Anteil an Objekten aus einer unendlichen Anzahl n von Objekten.

$$Pr(C_j) = \lim_{n\to\infty} \frac{n_j}{n} \tag{2.30}$$

Für eine große Anzahl n kann $Pr(C_j)$ approximiert werden durch: $Pr(C_j) = \frac{n_j}{n}$.

Die *a priori* Wahrscheinlichkeit wird aufgrund von Erfahrungen -hier durch das Lernen eines Klassifikators- in eine *a posteriori* Wahrscheinlichkeit umgerechnet. Die *a posteriori* Wahrscheinlichkeit $Pr(C_j|x)$ ist also die durch einen Klassifikator ermittelte Wahrscheinlichkeit, dass ein Objekt x der Klasse j angehört. $Pr(C_j|x)$ steht gemäß der Wahrscheinlichkeitstheorie im Zusammenhang mit verschiedenen Wahrscheinlichkeitsfunktionen. Dabei entspricht $p(x)$ der Verteilungsdichte für das Auftreten des Merkmalsmusters x unabhängig von der Klassenzugehörigkeit. $p(x)$ ist definiert als:

$$p(x) = \sum_j p(x \mid C_j) \cdot Pr(C_j) \tag{2.31}$$

Die in Gleichung (2.31) enthaltene klassenbedingte Verteilungsdichte $p(x|C_j)$ beschreibt das erwartete Auftreten des Merkmalsmusters x unter der Annahme, dass x zur Klasse C_j gehört. Die gemeinsame Verteilungsdichte $p(C_j,x)$ gibt Auskunft über die Wahrscheinlichkeit ein Objekt zu finden, das zu Klasse C_j gehört und das Merkmalsmuster x hat. Es bestehen folgende Zusammenhänge:

$$p(C_j, x) = Pr(C_j \mid x) \cdot p(x) \tag{2.32}$$

$$p(C_j, x) = p(x \mid C_j) \cdot Pr(C_j) \tag{2.33}$$

Dies lässt sich durch Umstellung in das BAYES-Theorem (Gl. (2.34) umwandeln:

$$Pr(C_j \mid x) = \frac{p(x \mid C_j) \cdot Pr(C_j)}{p(x)} \tag{2.34}$$

Die vorangehenden Gleichungen sind in Abbildung 2.14 graphisch veranschaulicht.

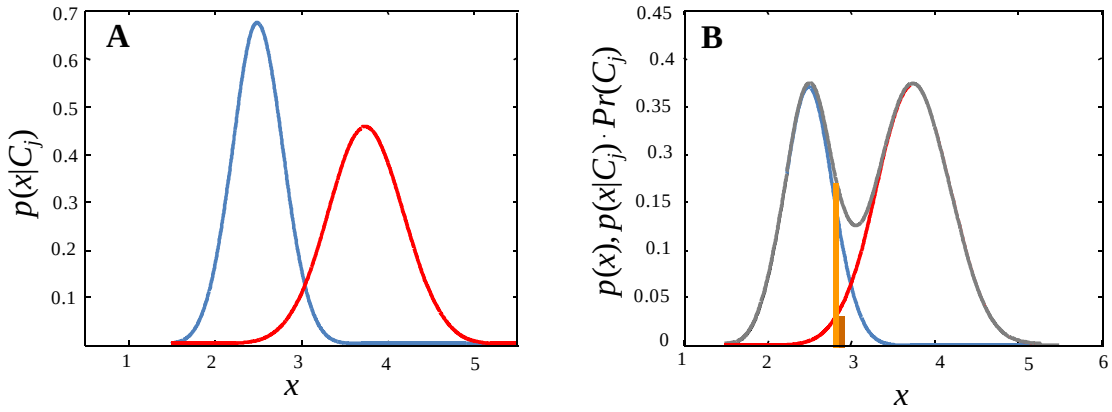


Abbildung 2.14: Skizze charakteristischer Verteilungsgrößen am Beispiel univariat-normalverteilter Daten. **A**: Klassenbedingte Verteilungsdichte für Klasse C_1 ($Pr(x|C_1)$) (blaue Linie) und Klasse C_2 ($Pr(x|C_2)$) (rote Linie). **B**: Gemeinsame Verteilungsdichten $p(C_j,x)$ (jeweils blaue und rote Line für Klasse 1 und 2) und $p(x)$ (graue Linie). Für die Berechnung der gemeinsamen Verteilungsdichten wurden folgende *a priori* Wahrscheinlichkeiten verwendet: $Pr(C_1)$ = 40% und $Pr(C_2)$ = 60% . Das Verhältnis von braunem zu orangem Balken beschreibt gemäß dem BAYES-Theorem die *a posteriori* Wahrscheinlichkeit $Pr(C_2|x=2.8)$, dass das Objekt des Musters x = 2.8 zur Klasse C_2 gehört. Wie man anhand der Graphik sehen kann, hat $Pr(C_2|x=2.8)$ einen relativ kleinen Wert ($Pr(C_2|x=2.8)$=14%). Es ist also wahrscheinlicher, dass ein Objekt des Musters x=2.8 zu Klasse C_1 (blaue Linie) gehört, als zu Klasse C_2 (rote Linie) ($Pr(C_1|x=2.8)$ =(1- $Pr(C_2|x=2.8)$=86%).

Das Ziel der BAYES-Entscheidungsregel (Gl. (2.35)) ist es, das Risiko von Falschklassifikationen zu minimieren. So wird ein unbekanntes Objekt der Klasse zugeordnet, für die die *a posteriori* Wahrscheinlichkeit $Pr(C_j \mid \mathrm{x})$ am größten ist.

Zur besseren Übersichtlichkeit wird in Gl. (2.35) angenommen, dass nur zwei Klassen C_1 und C_2 existieren.

$$f(x) = \begin{cases} C_1, \text{ falls } Pr(C_1 \mid \mathrm{x}) > Pr(C_2 \mid \mathrm{x}) \\ \\ C_2, \text{ falls } Pr(C_2 \mid \mathrm{x}) > Pr(C_1 \mid \mathrm{x}) \end{cases} \tag{2.35}$$

Die Minimierung der Fehlerwahrscheinlichkeit ist nicht immer das beste Kriterium, um die Kosten zu minimieren, die durch eine Fehlentscheidung entstehen. Eine Erweiterung der BAYES-Entscheidungsregel sind die kostenoptimalen Entscheidungsregeln [62]. Dabei wird eine Funktion verwendet, die verschiedene Arten der Fehlklassifikation unterschiedlich gewichtet je nach den Kosten die dabei entstehen würden. In dieser Arbeit wird die einfache BAYES-Entscheidungsregel für die Klassifikation der Bakterienstämme zugrunde gelegt.

Im Folgenden bezeichnet **X** (n x m) eine Matrix, die aus n Trainingsobjekten (hier Raman-Spektren) und m Variablen (hier gemessene Wellenzahlen) besteht. Die Objekte in **X** gehören k verschiedenen Klassen (hier Bakterienstämmen) an. g codiert die qualitativen Gruppen, so dass g_i dem Klassenlabel j des i-ten Objektes (hier Spektrums) entspricht. n_j steht für die Anzahl an Objekten, die zur Klasse j gehören. Die Wahrscheinlichkeit, dass ein neues Objekt **x** zur Klasse j gehört, wird mit $Pr(C_j|\mathbf{x})$ beschrieben. C bezieht sich auf die Klassenzugehörigkeit des Vektors **x**.

2.3.3.4 Klassifikationsalgorithmen

In den folgenden Kapiteln werden die für die Differenzierung des Bakteriendatensatzes verwendeten Klassifikationsalgorithmen vorgestellt.

2.3.3.4.1 "Partial Least Squares"-Diskriminanzanalyse (PLS-DA)

PLS-DA [63] ist eine der am häufigsten angewendeten Klassifikationsmethoden in der Chemometrik. Mit dieser Methode werden Objekte unter Verwendung einer „Partial Least Squares“ Regression [56,57] klassifiziert. Dabei wird zunächst eine Indikator Matrix **Y** (nxk)

erstellt, welche die Klassenindizes der Trainingsobjekte in **X** enthält. **Y** besteht nur aus den Werten 0 und 1. Gehört ein Objekt $\mathbf{x}_i$ der *j*-ten Klasse an, so hat **Y** an der Stelle **Y**(*i*, *j*) den Wert 1; alle anderen Werte in Zeile *i* sind 0. Das Klassifikationsmodell wird durch eine PLS-Regression von **Y** auf die Prediktor Matrix **X** trainiert. Wie in der multiplen linearen Regression (MLR) wird dabei ein lineares Modell der Form $\mathbf{Y}=\mathbf{X}\cdot\mathbf{B}+\mathbf{F}$ gebildet, wobei **B** die Matrix der Regressionskoeffizienten darstellt und **F** der Fehlermatrix entspricht. Im Gegensatz zur MLR werden jedoch bei der PLS-Regression die ursprünglichen Variablen aus **X** in neue latente Variablen, den PLS-Komponenten umgerechnet, die dann für die Regression verwendet werden. Die PLS-Regression hat gegenüber der MLR den Vorteil, dass durch Dimensionsreduktion auch für Daten mit hoher Multikollinearität eine lineare Regression gerechnet werden kann. Bei Verwendung der maximalen Anzahl an PLS-Komponenten A_{max} (Rang von **X**) entspricht die PLS-Regression der MLR. Die PLS ist eng verwandt mit der Hauptkomponentenanalyse (siehe Kapitel 2.3.2). Während bei der PCA nur für die Datenmatrix **X** latente Variablen gebildet werden, basiert die Berechnung der PLS-Komponenten sowohl auf den unabhängigen Variablen **X**, als auch auf den abhängigen Variablen **Y**. Zwischen den „Scores“ (**T**) der latenten Variablen von **X** und den „Scores“ (**U**) der latenten Variablen von **Y** wird ein Regressionsmodell erstellt. $\mathbf{U} = \hat{\mathbf{B}}\cdot\mathbf{T}$. $\hat{\mathbf{B}}$ ist eine Diagonalmatrix und enthält die Regressionskoeffizienten $\hat{\beta}_j$. Die „Scores“ in **T** und **U** werden so bestimmt, dass sie maximal miteinander korrelieren. Die entsprechenden **X**- und **Y**-„Loadings“ sind durch die Matrizen **P** und **Q** dargestellt. Diese Beziehungen sind in Abbildung 2.15 veranschaulicht.

PLS ist also eine Form der Dimensionsreduktion, bei der Richtungen im Datenraum von **X** gesucht werden, in denen **X** und **Y** maximal korreliert sind und gleichzeitig eine hohe Varianz vorhanden ist (vgl. PCA). Die Berechnung der PLS kann auf verschiedene Arten erfolgen. Eine gängige Methode zur Bestimmung von **T**, **P**, **U** und **Q** ist iterativ über den NIPALS (Nonlinear Iterative Partial Least Squares) Algorithmus [43]. Für die Berechnungen der Score-Matrizen **T** und **U** sind dabei zusätzlich gewichtete „Loadings“ erforderlich, die die Korrelationen zwischen **X**- und **Y**-Daten beschreiben. Diese gewichteten **X**- und **Y**-“Loadings“ sind durch die Matrizen **W** und **C** vertreten.

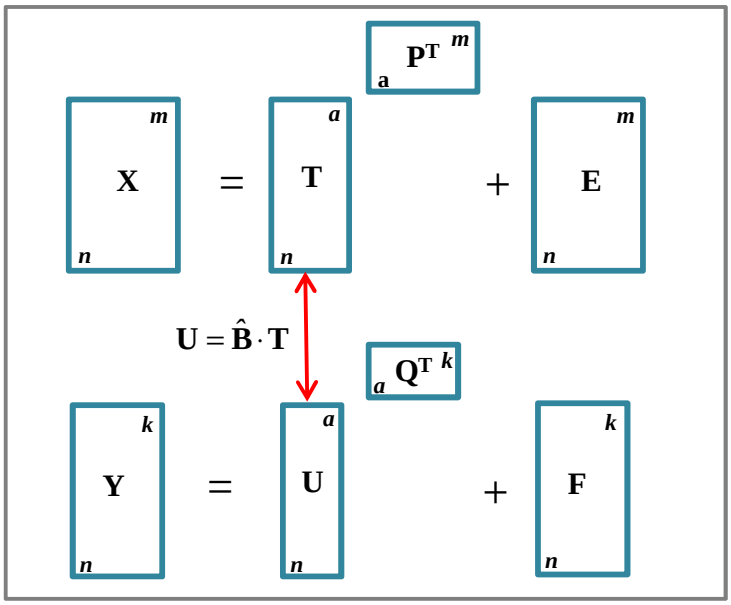


Abbildung 2.15: Graphische Veranschaulichung der PLS. Die Matrix **X** wird in eine Matrix **T** (die "Score"-Matrix) und eine Matrix $\mathbf{P}^T$ (die „Loadings“-Matrix) plus einer Fehlermatrix **E** zerlegt. Die Matrix **Y** wird in die Matrizen **U** (**Y**-„Scores“) und $\mathbf{Q}^T$ (**Y**-„Loadings“) und den Fehlerterm **F** zerlegt. Die „Scores“ **U** und **Q** werden iterativ so bestimmt, dass sie maximal miteinander korrelieren.

Im folgenden Algorithmus werden *a* PLS2-Komponenten bestimmt (PLS2 bedeutet im Gegensatz zu PLS1, dass **Y** aus mehreren Spalten besteht). *a* bezeichnet den Rang der Score-Matrix **T** und kann Werte zwischen 1 und A_{max} annehmen. Bei der PLS2 könnte als Anfangswert für die **Y**-“Scores“ ($\mathbf{u}_a$) irgendeine Spalte aus **Y** genommen werden. Normalerweise verwendet man aber den **y**-Vektor mit der größten euklidischen Norm $\max(\|\mathbf{Y}_j\|)$. Der NIPALS-Algorithmus (Algorithmus 2.2) wird also folgendermaßen initialisiert:

a=1

$\mathbf{X}_a$=**X** (spaltenzentriert)

$\mathbf{Y}_a$=**Y** (spaltenzentriert)

$\mathbf{u}_a = \max(\|\mathbf{Y}_j\|)$

Algorithmus 2.2: NIPALS-Algorithmus zur Berechnung der PLS2-Komponenten

1. Bestimme gewichtete $\mathbf{X}_a$ -"Loadings" (*a*-te Spalte von $\mathbf{W}_a$): $\mathbf{w}_a = \mathbf{X}_a^{\mathbf{T}} \cdot \mathbf{u}_a / (\mathbf{u}_a^{\mathbf{T}} \cdot \mathbf{u}_a)$
2. Skaliere $\mathbf{w}_a$ auf die Länge Eins: $\mathbf{w}_a = \mathbf{w}_a / || \mathbf{w}_a ||$
3. Berechne die $\mathbf{X}_a$ -"Scores" (*a*-te Spalte von $\mathbf{T}_a$): $\mathbf{t}_a = \mathbf{X}_a \cdot \mathbf{w}_a$
4. Bestimme gewichtete $\mathbf{Y}_a$-"Loadings" (*a*-te Spalte von $\mathbf{C}_a$): $\mathbf{c}_a = \mathbf{Y}_a^{\mathrm{T}} \cdot \mathbf{t}_a / (\mathbf{t}_a^{\mathbf{T}} \cdot \mathbf{t}_a)$
5. Skaliere $\mathbf{c}_a$ auf die Länge Eins: $\mathbf{c}_a = \mathbf{c}_a / || \mathbf{c}_a ||$
6. Berechne die $\mathbf{Y}_a$-"Scores" (*a*-te Spalte von $\mathbf{U}_a$): $\mathbf{u}_a = \mathbf{Y}_a^{\mathbf{T}} \cdot \mathbf{c}_a \cdot (\mathbf{c}_a^{\mathbf{T}} \cdot \mathbf{c}_a)$
7. Bei Konvergenz fahre mit Punkt 8 fort, andernfalls gehe zurück zu Punkt 1
8. Berechne $\mathbf{X}_a$-"Loadings" (*a*-te Spalte von $\mathbf{P}_a$): $\mathbf{p}_a = \mathbf{X}_a^{\mathbf{T}} \cdot \mathbf{t}_a \cdot (\mathbf{t}_a^{\mathbf{T}} \cdot \mathbf{t}_a)$
9. Berechne $\mathbf{Y}_a$-"Loadings" (*a*-te Spalte von $\mathbf{Q}_a$): $\mathbf{q}_a = \mathbf{Y}_a^{\mathbf{T}} \cdot \mathbf{u}_a \cdot (\mathbf{u}_a^{\mathbf{T}} \cdot \mathbf{u}_a)$
10. Regression von $\mathbf{u}_a$ auf $\mathbf{t}_a$: $\hat{\beta}_a = \mathbf{u}_a^{\mathbf{T}} \cdot \mathbf{t}_a / (\mathbf{t}_a^{\mathbf{T}} \cdot \mathbf{t}_a)$
11. Berechne Fehlermatrizen: $\mathbf{E}_a = \mathbf{X}_a - \mathbf{t}_a \cdot \mathbf{p}_a^{\mathbf{T}}$; $\mathbf{F}_a = \mathbf{Y}_a - \hat{\beta}_a \cdot \mathbf{t}_a \mathbf{c}_a^T$
12. $\mathbf{X}_{a+1} = \mathbf{E}_a$; $\mathbf{Y}_{a+1} = \mathbf{F}_a$
13. Zur Bestimmung weiterer PLS-Komponenten, beginne erneut bei Punkt 1 mit a=a+1

Neben der iterativen Bestimmung der PLS-Komponenten kann PLS auch durch mehrere Eigenwertprobleme definiert werden. Dies ist analog zur PCA, die sowohl durch einen NIPALS-Algorithmus als auch durch eine Eigenwertberechnung gelöst werden kann.
Für die Klassifikation eines neuen Objektes $\mathbf{x}_{\text{neu}}$ mittels PLS-DA wird auf den Trainingsdaten zunächst ein PLS-Regressionsmodell zwischen den zentrierten Daten **X** und **Y** erstellt:

$$\mathbf{Y} = \mathbf{X} \cdot \mathbf{B}_{\text{PLS}} + \mathbf{F}_{\text{PLS}} \tag{2.36}$$

Die Koeffizientenmatrix $\mathbf{B}_{PLS}$ wird geschätzt durch:

$$\hat{\mathbf{B}}_{PLS} = \mathbf{W} \cdot (\mathbf{P}^{\mathbf{T}} \cdot \mathbf{W})^{-1} \cdot \hat{\mathbf{B}} \cdot \mathbf{C}^{\mathbf{T}} \tag{2.37}$$

Dabei wurden **X** und **Y** durch „Scores“ und „Loadings“ des PLS-Modells ersetzt.
Für das neue Objekt $\mathbf{x}_{neu}$ wird $\hat{\mathbf{y}}$ folgendermaßen berechnet:

$$\hat{\mathbf{y}} = \bar{\mathbf{y}} + (\mathbf{x}_{neu} - \bar{\mathbf{x}}) \cdot \hat{\mathbf{B}}_{PLS} \tag{2.38}$$

$\mathbf{x}_{neu}$ wird anschließend derjenigen Klasse zugeordnet, für die der entsprechende Wert in $\hat{\mathbf{y}}$ maximal ist (der Wert, der 1 am nächsten ist).
Die Anzahl an PLS-Komponenten, die für die Regression verwendet werden, wird hier durch 50-fache Kreuzvalidierung bestimmt, wie es in Kapitel 2.3.2 für die Bestimmung der Anzahl an Hauptkomponenten beschrieben ist.
Es besteht eine enge Beziehung zwischen einer Klassifikation über lineare Regression (z. B. PLS-DA) und Fischer's Dikriminanzanalyse (LDA) (siehe Kapitel 2.3.3.4.2). So sind die beiden Klassifikationstechniken für Zwei-Klassen-Probleme äquivalent. Für die Analyse von Multi-Klassen Problemen hingegen erweist sich LDA einer Klassifikation mittels Regression in der Regel überlegen [60].

2.3.3.4.2 Lineare Diskriminanzanalyse (LDA)

Die lineare Diskriminanzanalyse, die 1936 von Fisher [64] begründet wurde, ist -wie die PLS-DA- ein Verfahren zum Auffinden linearer Entscheidungsgrenzen zwischen zwei oder mehreren Klassen. Der Algorithmus der LDA basiert auf der Annahme, dass die Klassen multivariat normalverteilt sind. Die Klassenmittelwerte $\boldsymbol{\mu}_j$ (Gl. (2.39)) und die "gepoolte" Varianz-Kovarianz-Matrix $\hat{\boldsymbol{\Sigma}}$ können leicht auf Basis der Trainingsdaten geschätzt werden. "Gepoolt" bedeutet in diesem Zusammenhang, dass die Varianz-Kovarianz-Matrix, die für alle Klassen als gleich angenommen wird, durch Aufsummieren der geschätzten Varianz-Kovarianz-Matrizen der einzelnen Klassen berechnet wird (Gl. (2.40)).

$$\hat{\boldsymbol{\mu}}_j = \sum_{g_i=j} \mathbf{x}_i \,/\, n_j \tag{2.39}$$

$$\hat{\boldsymbol{\Sigma}} = \sum_{j=1}^{k} \sum_{g_i=j} (\mathbf{x}_i - \hat{\boldsymbol{\mu}}_j) \cdot (\mathbf{x}_i - \hat{\boldsymbol{\mu}}_j)^{\mathrm{T}} \,/ (n-k) \tag{2.40}$$

Der Ausdruck $\sum_{g_i=j}$ steht für das Aufsummieren über alle Objekte *i* der Klasse *j*.

Eine neue Beobachtung $\mathbf{x}_{\text{neu}}$ wird der Klasse zugeordnet, für die die Diskriminanzfunktion $d_j(\mathbf{x})$ (Gl.(2.41)) maximal wird. Der erste Term von $d_j(\mathbf{x})$ entspricht der quadrierten Mahalanobis-Distanz $D(\mathbf{x}, \boldsymbol{\mu}_j)$ (siehe Gl. (2.28)) zwischen Objekt $\mathbf{x}$ und dem Zentroiden $\boldsymbol{\mu}_j$ der Klasse *j*. Der zweite Term ($\ln \Pi_j$) beinhaltet die *a priori* Wahrscheinlichkeit Π_j der Klasse *j*, die über $\Pi_j = n_j / n$ geschätzt wird. Das Maximieren der Funktion $d_j(\mathbf{x})$ ist äquivalent zum Maximieren der *a posteriori* Wahrscheinlichkeit $Pr(C_j|\mathbf{x})$, welche die Wahrscheinlichkeit angibt, dass ein neues Objekt $\mathbf{x}_{\text{neu}}$ zur Klasse *j* gehört. Unter Verwendung der bedingten Klassendichte $p(\mathbf{x}|C_j)$, kann die *a posteriori* Wahrscheinlichkeit $Pr(C_j|\mathbf{x})$ gemäß dem BAYES-Theorem (siehe Kapitel 2.3.3.3) berechnet werden (Gl. (2.43)).

$$d_j(\mathbf{x}) = -D(\mathbf{x}, \boldsymbol{\mu}_j) / 2 + \ln \Pi_j \tag{2.41}$$

$$p(\mathbf{x} \mid C_j) = \frac{1}{(2\Pi_j)^{m/2} \cdot |\boldsymbol{\Sigma}|^{1/2}} \cdot \mathrm{e}^{-D(\mathbf{x}, \boldsymbol{\mu}_j)/2} \tag{2.42}$$

$$Pr(C_j \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid C_j) \cdot \Pi_j}{\sum_{j=1}^{k} p(\mathbf{x} \mid C_j) \cdot \Pi_j} \tag{2.43}$$

Dabei bezeichnet $|\boldsymbol{\Sigma}|$ die Determinante der Varianz-Kovarianz-Matrix $\boldsymbol{\Sigma}$.

2.3.3.4.3 Quadratische Diskriminanzanalyse (QDA)

Der Algorithmus der QDA [60] ist dem Algorithmus der LDA sehr ähnlich. Jedoch wird bei der QDA im Gegensatz zur LDA eine separate Varianz-Kovarianz Matrix für jede einzelne Klasse berechnet. Dadurch erhält QDA quadratische Entscheidungsgrenzen. Die entsprechende Diskriminanzfunktion $d_j(\mathbf{x})$ lautet :

$$d_j(\mathbf{x}) = -\log\left|\boldsymbol{\Sigma}_j\right| - D(\mathbf{x},\boldsymbol{\mu}_j)/2 + \ln\Pi_j \tag{2.44}$$

Trotz der Tatsache, dass Daten aus der Praxis in der Regel nicht einer multivariaten Normalverteilung folgen, erweisen sich LDA und QDA als sehr leistungsfähig bei verschiedensten Klassifikationsproblemen. Außerdem zeichnen sich die Modelle meistens durch eine erstaunlich hohe Robustheit aus [60]. LDA schneidet in vielen Fällen besser ab als QDA, da weniger Parameter geschätzt werden müssen, bzw. die Schätzung einer „gepoolten" Varianz-Kovarianz Matrix bei wenigen Klassifikationsobjekten stabiler ist als die Schätzung separater Klassen-spezifischer Varianz-Kovarianz Matrizen [65]. In Situationen, in denen genügend Objekte verfügbar sind, um sinnvolle Schätzungen der Verteilungsparameter zu gewährleisten, kann sich die QDA als vorteilhaft erweisen. Dies ist besonders bei Vorliegen von nichtlinearen Klassifikationsproblemen der Fall.

2.3.3.4.4 "Gaussian Mixture" Diskriminanzanalyse (MDA)

a) Klassifikation mittels MDA

Die "Gaussian Mixture" Diskriminanzanalyse (MDA) [66] stellt eine weitere Variation des Prototyp Klassifikators LDA dar. Bei der MDA wird angenommen, dass jede Klasse aus mehreren normalverteilten Unterklassen besteht. Die klassenbedingten Verteilungsdichten $p(\mathbf{x}|C_j)$ werden durch eine multivariate Gauss´sche Mischungsverteilung (engl. Gaussian Mixture Model) beschrieben. Dadurch können nichtlineare Entscheidungsgrenzen entstehen. Da die zu einer Klasse gehörenden Unterklassen nicht notwendigerweise aneinander angrenzen müssen, können außerdem streuende Klassen-Strukturen modelliert werden. In der Originalversion der MDA von Hastie und Tibshirani wird angenommen, dass die Anzahl der

Unterklassen R_j für jede Klasse im Vorfeld der Analyse bekannt ist und dass die Varianz-Kovarianz Matrix für alle Klassen und Unterklassen identisch ist. Beide Annahmen wurden in nachfolgenden Arbeiten entschärft, was zu einer höheren Flexibilität der MDA führte [67,68]. Um die Anzahl der Parameter unter Kontrolle zu halten, wird hier eine Version der MDA verwendet, die die genannten Parameter konstant hält. Es wird also angenommen, dass jede Klasse dieselbe Anzahl an Unterklassen hat. Die Varianz-Kovarianz Matrix wird über alle Klassen und Unterklassen „gepoolt" (siehe Kapitel 2.3.3.4.2). Für die initiale Schätzung der Zentroiden der Unterklassen sowie der Ausgangs-Mischungswahrscheinlichkeiten $\hat{\pi}_{jr}$ wird eine *k*-Means Clusteranalyse durchgeführt. Dabei werden die Objekte im Vorfeld der Mischmodellbildung in die jeweiligen Unterklassen eingeteilt. $\hat{\pi}_{jr}$ entspricht der *a priori* Wahrscheinlichkeit für das Auftreten der jeweiligen Unterklasse *r*, unter der Voraussetzung, dass Klasse *j* gegeben ist. Die Schätzung von $\hat{\pi}_{jr}$ ergibt sich aus $\hat{\pi}_{jr} = \frac{n_{jr}}{n_j}, \sum_{r=1}^{R_j} \hat{\pi}_{jr} = 1$. Der *k*-Means Algorithmus durchläuft folgende Schritte:

Algorithmus 2.3: *k*-Means Algorithmus zur Initialisierung der MDA

1. Wähle zufällig aus den Datenpunkten einer Klasse *k* Clusterzentren
2. Ordne jedes Objekt dem ihm am nächsten liegenden Clusterzentrum zu (Berechnung der euklidischen Distanz)
3. Berechne für jeden Cluster die Clusterzentren neu
4. Falls sich die Zuordnung der Objekte ändert, fahre fort mit Schritt 2, ansonsten Abbruch

Die Gruppierung der Objekte durch den *k*-Means Algorithmus wird anschließend durch die Aufstellung der Mischungsverteilung optimiert. Der wesentliche Unterschied zwischen dem Clustering nach dem *k*-Means Algorithmus und dem Clustering mit Hilfe der Mischungsverteilung besteht darin, dass bei letzterem die Korrelationen zwischen den Variablen in den Abstandsberechnungen berücksichtigt werden. So basiert die Zuordnung eines Objektes zu einem Cluster beim *k*-Means Algorithmus auf der euklidischen Distanz

(siehe Kapitel 2.3.3.1.1), wohingegen bei der Gauss´schen Mischungsverteilung die Mahalanobis-Distanz (siehe Kapitel 2.3.3.1.2) das maßgebliche Distanzmaß darstellt. Die Schätzung der Parameter der unbekannten Mischungsverteilung (*a priori* Wahrscheinlichkeiten $\hat{\pi}_{jr}$, Zentroiden $\hat{\boldsymbol{\mu}}_{jr}$ und Varianz-Kovarianz Matrix $\hat{\boldsymbol{\Sigma}}$) erfolgt durch den „Expectation Maximization" Algorithmus (EM-Algorithmus), der auf einer Maximum-Likelihood Schätzung für die jeweiligen normalverteilten Unterklassen basiert. Die Maximum-Likelihood-Methode ist ein wichtiges Prinzip zur Berechnung von Schätzfunktionen für die Parameter einer Verteilung [69]. Die Schätzung erfolgt dabei auf Basis einer Stichprobe aus der Gesamtpopulation. Als Maximum-Likelihood-Schätzer wird ein Parameter-Wert für die zugrundeliegenden Funktion (Gauss´sche Mischfunktion) bezeichnet, der dazu führt, dass die gegebene Stichprobe mit größter Wahrscheinlichkeit durch die Funktion mit diesem speziellen Parameter generiert wird. Im Fall des hier verwendeten Mischungsmodells entspricht $gmf_j(\mathbf{x})$ (Gl.(2.45)) der normalverteilten klassenbedingten Verteilungsdichte für die einzelnen Klassen. Diese hängt von den Parametern $\hat{\pi}_{jr}$, $\hat{\boldsymbol{\mu}}_{jr}$ und $\hat{\boldsymbol{\Sigma}}$ ab.

$$gmf_j(\mathbf{x}) = p(\mathbf{x} \mid C_j) = \frac{1}{(2 \cdot \Pi_j)^{m/2} \cdot |\boldsymbol{\Sigma}|^{1/2}} \cdot \sum_{r=1}^{R_j} \pi_{jr} \cdot \mathrm{e}^{-D(\mathbf{x},\boldsymbol{\mu}_{jr})/2} \tag{2.45}$$

Im Gegensatz zum üblichen Vorgehen, bei dem für fixe Parameter ($\hat{\pi}_{jr}$, $\hat{\boldsymbol{\mu}}_{jr}$, $\hat{\boldsymbol{\Sigma}}$) die Dichte beliebiger Werte $\mathbf{x}_1,\ldots\mathbf{x}_n$ bestimmt wird, wird bei der Likelihood-Funktion L^{mix} für beobachtete und somit feste Realisationen $\mathbf{x}_1,\ldots\mathbf{x}_n$ die Dichte als Funktion der Parameter $\hat{\pi}_{jr}$, $\hat{\boldsymbol{\mu}}_{jr}$, $\hat{\boldsymbol{\Sigma}}$ betrachtet.

$$L^{mix}(\boldsymbol{\mu}_{jr}\boldsymbol{\Sigma},\pi_{jr}) = \sum_{i=1}^{n_j} gmf_{g_i}(\mathbf{x}_i,\boldsymbol{\mu}_{jr},\boldsymbol{\Sigma},\pi_{jr}) \tag{2.46}$$

Wird diese Funktion in Abhängigkeit von $\boldsymbol{\mu}_{jr}, \boldsymbol{\Sigma}$ und π_{jr} maximiert, so erhält man die Maximum-Likelihood-Schätzung. Es werden also die Werte von $\boldsymbol{\mu}_{jr}, \boldsymbol{\Sigma}$ und π_{jr} gesucht, bei denen die Stichprobenwerte $\mathbf{x}_1,\ldots\mathbf{x}_n$ die größte Dichte- bzw. Wahrscheinlichkeitsfunktion haben. Dieses Maximierungsproblem kann gelöst werden, indem man von Gl. (2.46) die partiellen ersten Ableitungen nach $\boldsymbol{\mu}_{jr}, \boldsymbol{\Sigma}$ und π_{jr} bildet, diese gleich Null setzt und nach den jeweiligen Parametern auflöst. Da dieses bei Dichtefunktionen mit komplizierten Exponenten-Ausdrücken -wie in diesem Beispiel- sehr aufwändig werden kann, wird häufig die logarithmierte Likelihood-Funktion ℓ^{mix} verwendet. Diese besitzt an derselben Stelle wie die nicht-logarithmierte Dichtefunktion ein Maximum und ist einfacher zu berechnen:

$$\ell^{mix}(\boldsymbol{\mu}_{jr}, \boldsymbol{\Sigma}, \pi_{jr}) = \sum_{i=1}^{n_j} \log\left(gmf_{g_i}(\mathbf{x_i}, \boldsymbol{\mu}_{jr}, \boldsymbol{\Sigma}, \pi_{jr})\right) \tag{2.47}$$

Um ℓ^{mix} zu maximieren, wird der EM-Algorithmus verwendet, der zwischen einem „Expectation“ (E)- und einem „Maximization“ (M)- Schritt alterniert [66]. Im E-Schritt wird jedem Trainingsobjekt ein Gewicht $\hat{p}(c_{jr} \mid \mathbf{x}, j)$ bezüglich der Zugehörigkeit zu jedem einzelnen Cluster zugeteilt. Dies basiert auf dem Likelihood der entsprechenden Mischungsverteilung. Das Gewicht $\hat{p}(c_{jr} \mid \mathbf{x}, j)$ beschreibt die Wahrscheinlichkeit, dass ein Trainingsobjekt $\mathbf{x}_i$ zu der Unterklasse *r* gehört unter der Voraussetzung, dass es zu der Klasse *j* gehört. (d.h. c_{jr} bezieht sich auf die betrachtete Unterklasse). Die Berechnung von $\hat{p}(c_{jr} \mid \mathbf{x}, j)$ basiert auf den initialen Parametern $\hat{\boldsymbol{\mu}}_{jr}, \hat{\boldsymbol{\Sigma}}$ und π_{jr}, die nach Zuordnung der Objekte durch den *k*-Means Algorithmus geschätzt wurden. Im M-Schritt werden die Parameter erneut geschätzt, wobei jedes Objekt proportional zu seinem Gewicht $\hat{p}(c_{jr} \mid \mathbf{x}, j)$ in die Berechnung eingeht. Die im M-Schritt gefundenen Parameter werden verwendet, um einen neuen E-Schritt zu beginnen. Dieser Prozess wird bis zur Konvergenz iteriert. Im

Folgenden bezeichnet Π_j die *a priori* Wahrscheinlichkeit für das Auftreten von Klasse *j*. Diese wird geschätzt durch $\Pi_j = n_j / n$. Der EM-Algorithmus durchläuft folgende Schritte:

Algorithmus 2.4: EM-Algorithmus zur Schätzung der Parameter normalverteilter Mischungsverteilungen

---

1. Schätze die initialen Parameter $\hat{\boldsymbol{\mu}}_{jr}$, $\hat{\pi}_{jr}$ and $\hat{\boldsymbol{\Sigma}}$ (*k*-Means Clustering)

2. „Expectation"-Schritt: Berechne Gewichte für jedes Objekt und jede Unterklasse:

$$\hat{p}(c_{jr} \mid \mathbf{x}, j) = \frac{\pi_{jr} \cdot \mathrm{e}^{-D(\mathbf{x}, \boldsymbol{\mu}_{jr})/2}}{\sum_{r=1}^{R_j} \pi_{jr} \cdot \mathrm{e}^{-D(\mathbf{x}, \boldsymbol{\mu}_{jr})/2}}$$

3. „Maximization"-Schritt: Schätze die Mischungswahrscheinlichkeiten, Mittelwerte und Varianz-Kovarianz-Matrix:

$$\hat{\pi}_{jr} \propto \sum_{g_i = j} p(c_{jr} \mid \mathbf{x}_i, j), \; \sum_{r=1}^{R_j} \hat{\pi}_{jr} = 1$$

$$\hat{\boldsymbol{\mu}}_{jr} = \frac{\sum_{g_i = j} \mathbf{x}_i \cdot p(c_{jr} \mid \mathbf{x}_i, j)}{\sum_{g_i = j} p(c_{jr} \mid \mathbf{x}_i, j)}$$

$$\hat{\boldsymbol{\Sigma}} = (1/n) \cdot \sum_{j=1}^{k} \sum_{g_i = j} \sum_{r=1}^{R_j} p(c_{jr} \mid \mathbf{x}_i, j) \cdot (\mathbf{x}_i - \boldsymbol{\mu}_{jr}) \cdot (\mathbf{x}_i - \boldsymbol{\mu}_{jr})^{\mathrm{T}}$$

Wiederhole Schritt 2 und 3 bis sich jedes $\hat{p}(c_{jr} \mid \mathbf{x}, j)$ nicht um mehr als 10^{-10} verändert

---

Die *a posteriori* Wahrscheinlichkeiten werden nach dem BAYES-Theorem berechnet:

$$Pr(\mathrm{C}_j \mid \mathrm{x}) = \frac{\Pi_j p(\mathbf{x} \mid C_j)}{p(\mathbf{x})} = \frac{\Pi_j \sum_{r=1}^{R_j} \pi_{jr} \mathrm{e}^{-D(\mathbf{x},\boldsymbol{\mu}_{jr})/2}}{\sum_{j=1}^{J} \Pi_j \sum_{r=1}^{R_j} \pi_{jr} \mathrm{e}^{-D(\mathbf{x},\boldsymbol{\mu}_{jr})/2}} \tag{2.48}$$

In Gl. (2.48) beschreibt $p(\mathbf{x}|C_j)$ die klassenbedingte Wahrscheinlichkeitsdichte und $p(\mathbf{x})$ ist die Wahrscheinlichkeitsdichte unabhängig von der Klassenzugehörigkeit. $p(\mathbf{x})$ dient hier als Normalisierungsfaktor und gewährleistet, dass die Summe aller *a posteriori* Wahrscheinlichkeiten gleich Eins wird. Ein neues Objekt $\mathbf{x}_{\mathrm{neu}}$ wird der Klasse mit der größten *a posteriori* Wahrscheinlichkeit zugeteilt.

b) Detektion von Vorhersageausreißern mittels MDA

Neben der eigentlichen Klassifikation der Spektren, d.h. der Zuteilung zu einem der im Trainingsdatensatz enthaltenen Bakterienstämme, ist es in diesem Projekt wichtig, unbekannte Bakterien, also Bakterien, die nicht im Trainingsdatensatz enthalten sind, als nicht klassifizierbar zu erkennen. In einem solchen Fall spricht man von der Erkennung von Vorhersageausreißern.

Bei den bisher beschriebenen parametrischen Methoden (LDA, QDA und MDA) kann die aus der Normal- bzw. Mischungsverteilung geschätzte unbedingte Wahrscheinlichkeitsdichte $p(\mathbf{x})$ (für MDA siehe Gl. (2.48)) direkt als quantitatives Maß für die Verschiedenheit eines Spektrums vom Trainingsdatensatz dienen, da angenommen wird, dass Vorhersageausreißer in Bereiche des Datenraums fallen, für die $p(\mathbf{x})$ einen kleinen Wert annimmt [70,71]. Alternativ dazu kann die klassenbedingte Wahrscheinlichkeitsdichte $p(\mathbf{x}|C_j)$ für die Erkennung der Vorhersageausreißer herangezogen werden. In diesem Fall geht ein Klassifikationsschritt der Ausreißererkennung voraus. In der praktischen Durchführung der Detektion von Vorhersageausreißern werden die unbedingten bzw. die bedingten Wahrscheinlichkeitsdichten ($p(\mathbf{x})$ bzw. $p(\mathbf{x}|C_j)$) der Trainingsdaten bestimmt und in fallender Reihenfolge geordnet. Im Fall von $p(\mathbf{x})$ wird empirisch ein Grenzwert s festgelegt, ab

welchem ein Spektrum als Vohersageausreißer definiert wird. Bei Verwendung von $p(\mathbf{x}|C_j)$ muss ein Grenzwert s_j für jede einzelne Klasse festgelegt werden. Wie hoch s bzw. s_j liegen, richtet sich nach der gewünschten Trennung zwischen Ausreißern und Nicht-Ausreißern. Für eine Irrtumswahrscheinlichkeit von z% (d.h. die geschätzte Wahrscheinlichkeit, dass Objekte fälschlicherweise als Vorhersageausreißer deklariert werden, ist z%) wird die z-te Perzentile der Wahrscheinlichkeitsdichten der Trainingsdaten berechnet. Dieser Wert dient als Grenzwert s bzw. s_j. Die Objekte mit einem kleineren Wert $p(\mathbf{x})$ bzw. $p(\mathbf{x}|C_j)$ als der Grenzwert, werden als Vorhersageausreißer definiert.

Für die Bestimmung von s wird in dieser Arbeit eine Kreuzvalidierung mit dem Trainingsdatensatz durchgeführt. Dabei wird jeder Stamm einmal aus dem Trainingsdatensatz entnommen und als Testdatensatz verwendet. Die Außreißererkennung wird mit verschiedenen Werten für die Irrtumswahrscheinlichkeit durchgeführt (z=5, 10, 15, 20, 25, 30). Dasselbe Kreuzvalidierungsschema erfolgt auf Artebene; d.h. alle Stämme einer Art werden bei jedem Kreuzvalidierungsschritt einmal als Testset entnommen. Je nachdem, wie sich Sensitivität (Anteil richtig erkannter Ausreißer) bzw. Spezifizität (Anteil richtig zugeordneter Nicht-Ausreißer) [72] in diesem Kreuzvalidierungsschritt bei verschiedenen Werten von z verhalten, werden z bzw. s vom Benutzer festgelegt.

Parametrische Methoden wie LDA, QDA und MDA weisen einige Vorteile gegenüber nicht-parametrischen Methoden wie *k*NN (siehe Kapitel 2.3.3.4.5) und SVMs (siehe Kapitel 2.3.3.4.6) auf. Da bei LDA, QDA und MDA die Daten mit Hilfe von multivariaten Normalverteilungen beschrieben werden, ist es leicht, die Vertrauenswürdigkeit jeder Vorhersage über *a posteriori* Wahrscheinlichkeiten abzuschätzen. Um zuverlässige Wahrscheinlichkeitswerte zu erhalten, sollten die Daten dabei näherungsweise multivariat normalverteilt sein. Ein weiterer Vorteil der parametrischen Methoden besteht darin, dass Vorhersageausreißer, d.h. Spektren, die sich stark von den Spektren des Trainingsdatensatzes unterscheiden, leicht mit Hilfe der Verteilungsdichten $p(\mathbf{x})$ bzw. $p(\mathbf{x}|C_j)$ erkannt werden können. Dazu kommen eine hohe Robustheit der Modelle sowie eine leichte Interpretierbarkeit. Beispielsweise kann der Einfluss einzelner Variablen auf die Klassifizierung leicht aus einem LDA-Modell extrahiert werden. Ein besonderer Vorteil von MDA im Vergleich zu den anderen verteilungsbasierten Modellen ist, dass auch stark heterogene und im Datenraum streuende Klassen modelliert werden können. Die in der MDA

enthaltene Clusteranalyse bietet zusätzliche Informationen über die Datenstruktur jeder einzelnen Klasse und erhöht dadurch die Interpretierbarkeit des Modells (siehe Kapitel 2.3.5.1). Durch das Schätzen einer "gepoolten" Varianz-Kovarianz-Matrix erweist sich die MDA im Gegensatz zur QDA auch dann als stabil, wenn nur wenige Objekte pro Klasse vorhanden sind.

2.3.3.4.5 k-Nächste-Nachbarn Klassifizierer (kNN)

Die Klassifikationstechniken, die in den vorangehenden Kapiteln vorgestellt wurden, basieren auf der Annahme normalverteilter Daten. Keine Verteilungsannahmen sind dagegen für den im Folgenden beschriebenen *k*-Nächste-Nachbarn Klassifizierer notwendig, weswegen er als nicht-parametrischer Klassifikationsalgorithmus bezeichnet wird. Aufgrund der Einfachheit des Verfahrens, wird *k*NN auch *lazy learning* („faules Lernen“) genannt. Die Klassifikation eines neuen Objektes $\mathbf{x}_{neu}$ erfolgt auf der Basis einer Mehrheitsentscheidung seiner *k* nächsten Nachbarn. So wird $\mathbf{x}_{neu}$ derjenigen Klasse zugeordnet, zu der die meisten Objekte in seiner Nachbarschaft gehören. Die Anzahl der dabei in Betracht gezogenen Nachbarschaftsobjekte ist gleich *k*. In Abhängigkeit der gewählten Anzahl an nächsten Nachbarn bildet *k*NN hoch flexible (kleines *k*) oder auch weniger flexible (großes *k*) Entscheidungsgrenzen für die Klassifikation. Das für die Berechnung verwendete Abstandsmaß ist hier die euklidische Distanz. In Abbildung 2.16 ist eine Klassifikation mittels *k*NN dargestellt.

Trotz einfacher Durchführbarkeit, liefert *k*NN sehr gute Klassifikationsergebnisse für verschiedenste Datensätze aus der Praxis. Basierend auf dem „Nächste-Nachbarn“-Prinzip wurden zudem einige effektive Verfahren zur Erkennung von Vorhersageausreißern entwickelt [73]. Da bei dem *k*NN-Algorithmus *a posteriori* Wahrscheinlichkeiten nicht auf direktem Weg zugänglich sind, wurden verschiedene Techniken vorgeschlagen, um diese zu schätzen. Eine gängige Vorgehensweise ist die Kombinationen von *k*NN mit dem „Bootstrap“-Verfahren [74,75].

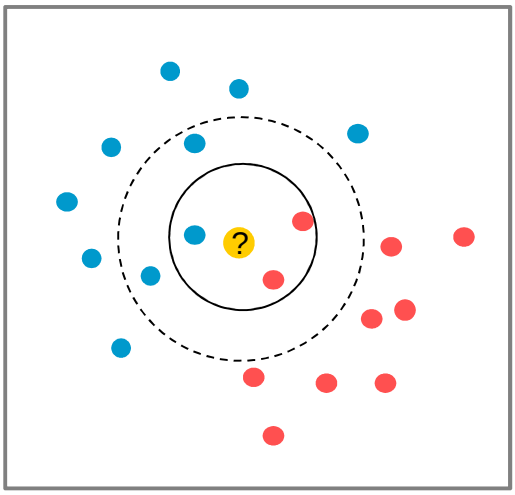


Abbildung 2.16: *k*NN Klassifikation mit *k*=3 (durchgezogene Linie) bzw. *k*=5 (gestrichelte Linie) Nachbarn. Bei Verwendung 3 nächster Nachbarn wird das neue Objekt ? der roten Klasse zugeteilt. Bei Verwendung 5 nächster Nachbarn wird ? der blauen Klasse zugeordnet.

2.3.3.4.6 *„Support Vector Machines“* (*SVMs*)

a) Klassifikation mittels SVMs

SVMs, die wie *k*NN zu den nicht-parametrischen Lernmethoden zählen, wurden im Jahr 1974 von Vapnik entwickelt und können sowohl zur Klassifikation als auch zur Regression eingesetzt werden [76]. In dem Fall der Klassifikation wird versucht, eine Hyperebene zu finden, für die die Trennspanne (engl. Margin) zwischen zwei Klassen maximal wird. Dabei wird der Abstand der Hyperebene zu den am nächsten liegenden Punkten aus beiden Klassen maximiert (siehe Abbildung 2.17A). Die Punkte, die zur Hyperebene am nächsten liegen, nennt man „Support Vectors“ („tragende Vektoren“). Sie allein bestimmen über die Lage der Hyperebene. Für den Fall, dass die Klassen nicht durch eine lineare Entscheidungsgrenze getrennt werden können, wurde ein "Soft Margin"-Algorithmus [77] entwickelt, der es erlaubt, dass einige Punkte ξ_i auf der „falschen“ Seite der Trennebene liegen (siehe

Abbildung 2.17B). Das Ausmaß dieser Fehlklassifikationen wird durch den benutzerdefinierten Parameter *C* reguliert. Die Entstehung der Hyperebene zwischen zwei Klassen ist in Abbildung 2.17 veranschaulicht.

Mathematisch ist die Hyperebene, die als Entscheidungsfunktion $\hat{f}(\mathbf{x})$ fungiert, gegeben durch den Normalenvektor **w** und die Verschiebung *b*.

$$\hat{f}(x) = \mathbf{w}^{\mathrm{T}} \cdot \mathbf{x} + b \tag{2.49}$$

Je nach Vorzeichen von $\hat{f}(\mathbf{x})$ wird ein neues Objekt einer der beiden Klassen zugeordnet.

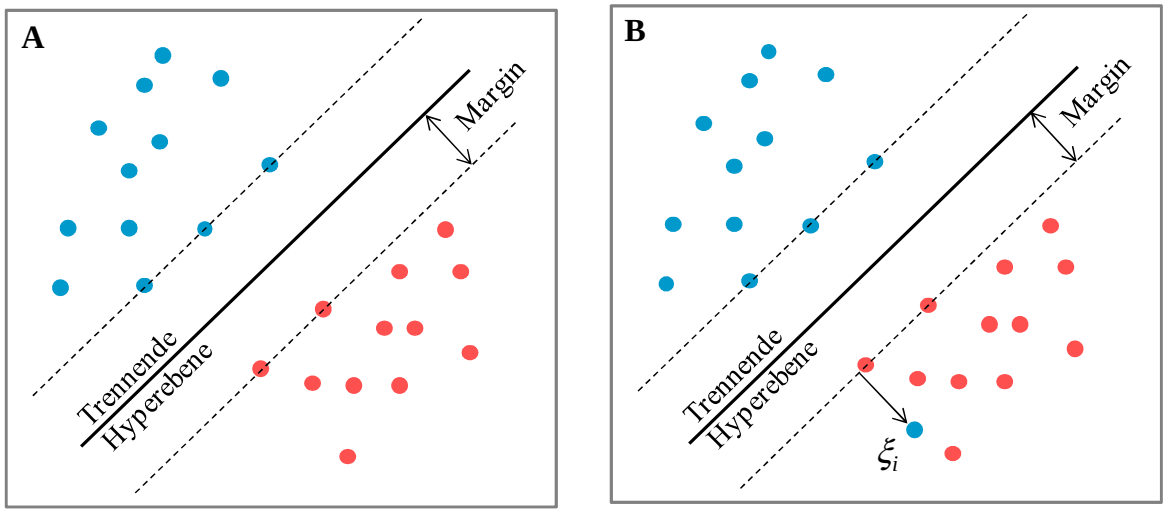


Abbildung 2.17A: Bildung der maximal trennenden Hyperebene durch SVMs im linear trennbaren Fall. Die auf der durchbrochenen Linie liegenden Punkte sind der Hyperebene am nächsten und entsprechen den „Support Vectors". Sie bestimmen über die Lage der Hyperebene. In **Abbildung 2.17B** lassen sich die Klassen nicht durch eine Hyperebene trennen. Hier wird ein „Soft Margin Classifier“ verwendet, bei dem das Ausmaß von Mißklassifikationen ξ_i durch den Parameter *C* reguliert wird.

Bei der Berechnung der Hyperebene wird die quadratische Norm des Normalenvektors **w** ($\|\mathbf{w}\|^2$) minimiert, wobei im linear nicht trennbaren Fall dieser Term durch den Missklassifikationsterm $C \cdot \sum_{i=1}^{n} \xi_i$ ergänzt wird. So erhält man folgendes quadratische Optimierungsproblem:

$$\varsigma(\mathbf{w},b) = \frac{1}{2} \cdot \|\mathbf{w}\|^2 + C \cdot \sum_{i=1}^{n} \xi_i \rightarrow \min! \tag{2.50}$$

mit der Nebenbedingung: $g_i \cdot (\mathbf{w}^{\mathrm{T}} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i,\ \xi_i \geq 0$

Dabei beschreibt g_i die Klasse des Trainingsobjektes *i*. Da es sich um ein Zwei-Klassenproblem handelt, gilt: $g_i \in \{+1,-1\}$. Die Lösung des Optimierungsproblems wird mit Hilfe der Lagrange-Multiplikatoren hergeleitet [78].

Die hohe Popularität der SVMs ist hauptsächlich auf ihre Fähigkeit zurückzuführen, komplexe, nichtlineare Modelle bilden zu können. Für diesen Zweck wird eine nichtlineare Funktion $\phi(\mathbf{x})$ (in der Regel eine Kernel-Funktion) eingeführt, um die Datensatz-Objekte in einen höher-dimensionalen Raum zu projizieren. Die optimal trennende Hyperebene wird dann in diesem erweiterten Raum konstruiert, was zu nichtlinearen Entscheidungsgrenzen im ursprünglichen Datenraum führt. In der vorliegenden Arbeit wird dafür ein RBF-Kernel (RBF: radiale Basisfunktionen) verwendet, der folgendermaßen definiert ist:

$$K(x_i, x_j) = \mathrm{e}^{-\gamma \|x_i - x_j\|^2} \text{ für } \gamma > 0 \tag{2.51}$$

Dabei stellen $\mathbf{x}_i$ und $\mathbf{x}_j$ zwei Trainingsobjekte des Datensatzes dar. Die radiale Spannweite γ muss vom Benutzer definiert werden.

SVMs wurden ursprünglich für Zwei-Klassen-Probleme entwickelt. Mit Hilfe unterschiedlicher Techniken (z. B. Binarisierung, siehe Kapitel 2.3.3.5), können sie auf Multi-Klassen-Klassifikationen erweitert werden.

Während bei der SVM-Klassifikation lange Zeit nur Klassenlabel vorhergesagt werden konnten (keine Angabe von *a posteriori* Wahrscheinlichkeiten), veröffentlichten Platt et. al [29] im Jahr 2000 eine Methode zur Schätzung von *a posteriori* Wahrscheinlichkeiten im Zwei-Klassen-Fall, die von Lin et al. [79] weiterentwickelt wurde und in der heutigen Implementierung der LIBSVM enthalten ist. Die Funktion nach Lin et al. lautet:

$$Pr(C_j|\mathbf{x}) = \frac{1}{1+\mathrm{e}^{A\cdot\hat{f}+B}} \qquad 2.52$$

Dabei werden *A* und *B* durch die Minimierung der negativen Log-Likelihood Funktion mit bekannten Trainingsdaten und deren Entscheidungsfunktionswerten $\hat{f}$ geschätzt. *A posteriori* Wahrscheinlichkeiten für den Multi-Klassenfall können durch Binarisierung und Umrechnung der einzelnen Zwei-Klassen-Wahrscheinlichkeiten in Multi-Klassen *a posteriori* Wahrscheinlichkeiten erhalten werden (siehe Kapitel 2.3.3.5.1).

b) Detektion von Vorhersageausreißern mittels SVMs

Die Erkennung von Vorhersageausreißern ist mit Hilfe einer Ein-Klassen-SVM (engl. „One-Class SVM“) möglich [80]. Dabei wird angenommen, dass ungewöhnliche Objekte in einen Bereich des Datenraums fallen, der nicht von den Trainingsdaten abgedeckt wird. Analog zur Zwei-Klassen-Klassifikation, werden die Daten bei der Ein-Klassen-SVM mit Hilfe einer geeigneten Funktion $\phi(\mathbf{x})$(hier RBF-Kernel) in einen höherdimensionalen Variablenraum projiziert. Es wird eine Hyperebene generiert, die die Trainingsdaten von ungewöhnlichen, neuen Objekten (Vorhersageausreißer) trennen soll. Man erhält wiederum eine Entscheidungsfunktion $\hat{f}(x)$, mit der je nach Vorzeichen entschieden wird, ob ein Objekt den Trainingsdaten ähnlich ist (+) oder ob es als Vorhesageausreißer deklariert wird (-):

$$\hat{f}(\mathbf{x}) = \mathrm{sgn}(\mathbf{w}\cdot\phi(\mathbf{x}) - \rho) \qquad (2.53)$$

Die Parameter $\mathbf{w}$ und ρ der Hyperebene werden erhalten, indem folgendes Optimierungsproblem gelöst wird:

$$\varsigma(\mathbf{w},\rho) = \frac{1}{2}\cdot ||\mathbf{w}||^2 + \frac{1}{\nu_n}\cdot\sum_{i=1}^{n}\xi_i - \rho \rightarrow \min! \tag{2.54}$$

Mit der Nebenbedingung $\mathbf{w}\cdot\phi(\mathbf{x}_i) \geq \rho - \xi_i$ und $\xi_i \geq 0$

Die Variable ξ_i beschreibt -wie im Zwei-Klassen Fall- Fehlzuordnungen. Das sind Objekte des Trainingsdatensatzes die auf der Seite der Hyperebene liegen, die den Vorhersageausreißern zugeordnet wird.

Anstelle des Parameters *C* im Zwei-Klassen-Fall, wird bei der Ein-Klassen-SVM der Parameter ν vom Benutzer definiert. ν reguliert den Anteil der zu detektierenden Vorhersageausreißer. Während bei kleinem ν nur wenige Objekte als Vorhersageausreißer erkannt werden, nimmt die Anzahl der zu detektierenden Vorhersageausreißer mit steigendem ν zu. Bei großen Werten von ν muss in Kauf genommen werden, dass auch mehrere Trainingsobjekte auf der Seite der Hyperebene liegen, die den Vorhersageausreißern zugeordnet wird. Mit steigendem ν wächst also auch der Anteil an falsch positiven Ausreißern. Der Parameter ν der Ein-Klassen-SVM entspricht dem Parameter z (Irrtumswahscheinlichkeit) bei der Ausreißererkennung mittels MDA (siehe Kapitel 2.3.3.4.4).
Ein Vorteil der SVMs ist die gute Vorhersagegenauigkeit bei verschiedensten Arten von praxisrelevanten Klassifikationsproblemen. Dabei sind keine Verteilungsannahmen für die Klassifikation nötig. Ein Nachteil der SVMs ist ihre mathematische Komplexität, was eine schlechte Interpretierbarkeit mit sich bringt, weswegen SVMs zu den "Blackbox"-Modellen gezählt werden.

2.3.3.5 Paarweise Klassifikation (PK)

Einige Klassifikationsalgorithmen, wie SVMs, wurden speziell für binäre Klassifikationsprobleme entwickelt. Dabei werden zu klassifizierende Datenpunkte einer von zwei Klassen zugeordnet. Die meisten Datensätze aus der Praxis weisen allerdings mehr als zwei Klassen auf. Derartige Probleme können für SVMs mit Hilfe einer Binarisierung gelöst werden, bei der ein Multiklassenproblem in eine Serie von Zweiklassenproblemen umgewandelt wird. Die daraus erhaltenen Klassifikationsergebnisse werden anschließend wieder in eine Multiklassen-Zuordnung umgerechnet. Bei Beschäftigung mit der Literatur über Klassifikation fällt auf, dass für Klassifikationsmethoden, die eine Binarisierung nicht "nötig haben" -d.h. Methoden die aufgrund ihres Algorithmus Multiklassenprobleme direkt lösen können- selten Binarisierungen verwendet werden. Binarisierung kann in bestimmten Fällen jedoch zu einem deutlichen Gewinn an Vorhersagekraft führen [81]. Insbesondere lineare Klassifizierer sind mit Hilfe der Binarisierung in der Lage nichtlineare Eigenschaften anzunehmen. Da in dieser Arbeit nichtlineare Datenstrukturen analysiert werden, soll untersucht werden, ob sich die Kombination verschiedener Klassifikationsmethoden mit Techniken der Binarisierung positiv auf die Wiedererkennungsrate auswirkt. Dabei gibt es unterschiedliche Herangehensweisen sowohl bezüglich der Art der Binarisierung als auch hinsichtlich der Zusammenfassung der binären Klassifikationsergebnisse [82,83].

2.3.3.5.1 Methoden der Binarisierung und Multiklassenzuordnung

Ein bekanntes Verfahren zur Binarisierung ist die sogenannte "One-Against-All" bzw. "One-Against-the Rest" Methode [82,83]. Diese wandelt ein Multiklassenproblem mit k Klassen in k binäre Probleme um, wobei jeweils eine Klasse des Datensatzes als Klasse 1 und alle restlichen Daten als Klasse 2 definiert sind. Auf Basis dieser Einteilung wird ein Klassifikationsmodell erstellt. Dies wird so oft wiederholt bis jede Klasse einmal gegen die restlichen Daten klassifiziert wurde. Ein neues Objekt $\mathbf{x}_{neu}$ kann am Ende derjenigen Klasse zugeordnet werden, für die die *a posteriori* Wahrscheinlichkeit (bzw. die Entscheidungsfunktion bei SVMs) den größten Wert hat [82].
Alternativ zur "One-Against-All" Methode schlugen Knerr und Friedman die "One-Against-One" Methode vor [81,84]. Dabei werden zwei Klassen aus dem Datensatz herausgegriffen

und ein Klassifikationsmodell wird erstellt. Dies wird sooft wiederholt bis für jede Zweierkombination einmal ein Klassifikationsmodell vorhanden ist. So werden für *k* Klassen $k \cdot (k\text{-}1)/2$ Modelle trainiert. Die einfachste Art, aus diesen Zwei-Klassen-Zuordnungen ein Multi-Klassen-Ergebnis zu erhalten, ist das "Major Vote"-Schema, welche auch "Max Wins rule" genannt wird und von Friedman [81] vorgeschlagen wurde. Beim "Major Voting" wird ein zu klassifizierendes unbekanntes Objekt derjenigen Klasse zugeordnet, zu der es bei den paarweisen Klassifikationen am häufigsten zugeteilt wurde. Als Weiterentwicklung des "Major Vote"- Schemas veröffentlichten Hastie und Tibshirani den "Pairwise Coupling" Algorithmus [85]. Bei diesem werden die Wahrscheinlichkeiten, mit denen ein Objekt in den paarweisen Klassifikationen einer bestimmten Klasse zugeteilt wird, in eine Gesamtwahrscheinlichkeit für die Multi-Klassenzugehörigkeit umgerechnet. Hastie und Tibshirani argumentierten, dass durch das Einbeziehen der *a posteriori* Wahrscheinlichkeiten der Zwei-Klassen-Entscheidungen die Wiedererkennungsrate im Vergleich zum einfachen "Major Vote"-Schema verbessert werden kann. Im Folgenden wird der Algorithmus des „Pairwise Coupling“ beschrieben. Dabei entspricht δ_{ij} der von einem Klassifizierer geschätzten *a posteriori* Wahrscheinlichkeit, dass beim paarweisen Vergleich von zwei Klassen *i* und *j* ein neues Objekt $\mathbf{x}_{\text{neu}}$ zur Klasse *i* gehört ($\delta_{ij} = Pr(C_i|C_i \text{ oder } C_j)$). Die dem geschätzten Wert δ_{ij} entsprechende „wahre“ *a posteriori* Wahrscheinlichkeit wird mit ω_{ij} bezeichnet. Gesucht wird nun die auf den Multi-Klassen-Fall bezogene a *posteriori* Wahrscheinlichkeit ($p_i = Pr(C_i|\mathbf{x})$ für $i = 1,\ldots k$), dass das Objekt $\mathbf{x}_{\text{neu}}$ zur Klasse *i* gehört. Es gilt:

$$\omega_{ij} = \frac{p_i}{p_i + p_j} \tag{2.55}$$

Zur Lösung dieses Problems schlagen Hastie und Tibshirani vor, die Kullback-Leibler (KL) Distanz $KL(\mathbf{p})$ zwischen δ_{ij} und ω_{ij} (Gl. (2.56)) zu minimieren:

$$KL(\mathbf{p}) = \sum_{i<j} n_{ij} \cdot \left[\delta_{ij} \cdot \log \frac{\delta_{ij}}{\omega_{ij}} + (1-\delta_{ij}) \cdot \log \frac{1-\delta_{ij}}{1-\omega_{ij}} \right] \tag{2.56}$$

n_{ij} beschreibt dabei die Anzahl der in *i* und *j* enthaltenen Trainingsobjekte. Der Vektor **p** enthält die zu schätzenden *a posteriori* Wahrscheinlichkeiten eines zu klassifizierenden Objekts für alle Klassen und wird durch Minimieren der Funktion (2.56) erhalten. Dies kann durch den folgenden Algorithmus erreicht werden.

Algorithmus 2.5: Pairwise Coupling

---

1. Initialisiere $\hat{p}_i$ und die entsprechenden Werte $\hat{\omega}_{ij}$ zufällig
2. Wiederhole für *i* = (1,2,…,*k*,1,…) bis zur Konvergenz:

   $$\hat{p}_i \leftarrow \hat{p}_i \cdot \frac{\sum_{j \neq i} n_{ij} \cdot \delta_{ij}}{\sum_{j \neq i} n_{ij} \cdot \hat{\omega}_{ij}}$$

   Berechne $\hat{\omega}_{ij}$ neu: $\hat{\omega}_{ij} = \frac{\hat{p}_i}{\hat{p}_i + \hat{p}_j}$
3. Normalisiere $\hat{\mathbf{p}}$: $\hat{\mathbf{p}} \leftarrow \hat{\mathbf{p}} / \sum \hat{p}_i$

Neben dem „Pairwise Coupling“-Algorithmus wurden weitere Methoden zur Kopplung der Wahrscheinlichkeiten aus binären Klassifikationsmodellen veröffentlicht [86]. In diesem Zusammenhang ist der Algorithmus nach Price, Knerr, Personnaz und Dreyfus [87] zu nennen, der sehr einfach und leicht zu implementieren ist. Dieser wird hier mit PKPD gekennzeichnet.

Price et al. gehen von folgender Annahme aus:

$$\left(\sum_{j:\, j \neq i} Pr(C_i \text{ oder } C_j | \mathbf{x}) \right) - (k-2) \cdot Pr(C_i | \mathbf{x}) = \sum_{j=1}^{k} Pr(C_j | \mathbf{x}) = 1 \tag{2.57}$$

Unter Verwendung von:

$$\delta_{ij} \approx \omega_{ij} = \frac{Pr(C_i | \mathbf{x})}{Pr(C_i \text{ oder } C_j | \mathbf{x})} \tag{2.58}$$

erhält man:

$$\hat{p}_i = \frac{1}{\sum_{j:\, j \neq i} \frac{1}{\delta_{ij}} - (k-2)} \tag{2.59}$$

Wu et al. schlugen einen Algorithmus vor, der in Kombination mit SVMs bei verschiedenen Datensätzen stabilere Ergebnisse liefert als „Paiwise Coupling“ und „Major Voting“ [86]. Dieser Algorithmus, der eine Weiterentwicklung des Algorithmus nach Refregier und Vallet [88] darstellt, wird als Standardmethode in der Toolbox LIBSVM für die Berechnung von Multi-Klassenwahrscheinlichkeiten für SVMs verwendet [89]. Folgender Term wird bei dieser Methode minimiert:

$$\zeta(p) = \sum_{i=1}^{k} \sum_{j:\, j \neq i} (\delta_{ji} \cdot p_i - \delta_{ij} \cdot p_j)^2 \rightarrow \min! \tag{2.60}$$

unter der Voraussetzung: $\sum_{i=1}^{k} p_i = 1,\ p_i \geq 0, \forall i$.

Es gibt mehrere Möglichkeiten, dieses quadratische Optimierungsproblem zu lösen [86]. Eine iterative Lösung liefert folgender Algorithmus:

Algorithmus 2.6: Berechnung von Multiklassenwahrscheinlichkeiten nach Wu et al. (gemäß Algorithmus 2 in [86])

4. Initialisiere $\hat{p}_i$, so dass $\hat{p}_i \geq 0, \forall i$ und $\sum_{i=1}^{k} \hat{p}_i = 1$
5. Wiederhole für $i = (1,2,\ldots,k,1,\ldots)$ bis Ausdruck (2.60) erfüllt ist

$$\hat{p}_i \leftarrow \frac{1}{Q_{ii}} \left[- \sum_{j:\, j \neq i} Q_{ij} \cdot \hat{p}_j + \hat{\mathbf{p}}^{\mathrm{T}} \cdot Q \cdot \hat{\mathbf{p}} \right]$$

6. Normalisiere $\hat{\mathbf{p}}$: $\hat{\mathbf{p}} \leftarrow \hat{\mathbf{p}} / \sum \hat{p}_i$

Dabei gilt:

$$Q_{ij} = \begin{cases} \sum_{o:o \neq i} \delta_{oi}^2, \text{wenn } i = j \\ -\delta_{ji} \cdot \delta_{ij}, \text{wenn } i \neq j \end{cases} \tag{2.61}$$

2.3.3.5.2 Auswirkung der Binarisierung auf die Klassifikation

In einigen Studien wurde eine Überlegenheit der "One-Against-One"-Methode gegenüber der "One-Against-All"-Methode bei der Klassifikation mittels SVMs deutlich [82]. Auch in dieser Arbeit wird der "One-Against-One" Methode der Vorzug gegeben, was folgenden Hintergrund hat. Durch die Binarisierung soll Flexibilität vor allem bei der Anwendung parametrischer Methoden in das Klassifikationsmodell eingeführt werden, da -wie in Kapitel 3.4.1.3.1 gezeigt- Nichlinearität sich bei der Klassifikation der Raman-Spektren als

vorteilhaft erweist. Für diesen Zweck ist speziell die "One-Against-One" geeignet. Dies lässt sich einfach am Beispiel der linearen Diskriminanzanalyse (LDA) erklären. Bei der LDA wird angenommen, dass alle Klassen die gleiche Varianz-Kovarianz-Matrix haben. Bei der paarweisen Klassifizierung mit der "One-Against-One" Methode gilt diese Annahme nur für jedes Klassenpaar, wohingegen bei der „One-Against-All“-Methode der gesamte Datensatz in die Berechnung der Varianz-Kovarianz-Matrix eingeht. Für den Fall, dass mehr als zwei Klassen vorhanden sind, führt dies bei der „One-Against-One“-Methode zu nichtlinearen Entscheidungsgrenzen, nachdem alle paarweisen Vorhersagen kombiniert wurden.
Man kann es als Nachteil der Binarisierung bewerten, dass jeder der binären Klassifizier eine Entscheidung zwischen zwei Klassen treffen muss, selbst wenn das neue Objekt zu keiner der zwei Klassen gehört. Dadurch kommt es gezwungenermaßen zu Falschklassifizierungen. Grundsätzlich könnte es vorkommen, dass auf diese Weise beim „Major Voting“ eine falsche Klasse die meisten Stimmen erhält. Jedoch geht man davon aus, dass die anderen Klassifizierer, die die korrekte Klasse des neuen Objekts enthalten, diese Falschklassifizierungen wieder wettmachen. Auch beim „Pairwise Coupling“ werden viele nicht relevante *a posteriori* Wahrscheinlichkeiten der paarweisen Klassifikationen in die Berechnung der Multi-Klassen Wahrscheinlichkeiten eingerechnet. Um dem entgegenzuwirken wurden einige Weiterentwicklungen des "Pairwise Coupling"-Verfahrens veröffentlicht, die dies korrigieren sollten [90,91].

2.3.4 Bewertungs- und Auswerteverfahren

2.3.4.1 Validierung von Klassifikationsmodellen

Bei der Klassifikation wird mit Hilfe von Trainingsdaten ein Modell erstellt. Es stellen sich dabei vor allem zwei Fragen:

a. Welches sind die optimalen „Tuning“-Parameter für das Modell?
b. Welche Vorhersagegenauigkeit ist nach der Modellbildung zu erwarten?

Wie bereits in den vorangehenden Kapiteln erwähnt, werden in dieser Arbeit unter „Tuning“-Parameter diejenigen Modellparameter verstanden, die vom Benutzer festzulegen sind. Das sind beispielsweise die Anzahl der latenten Variablen für PLS-DA, Anzahl der PCs für LDA, MDA, QDA und *k*NN, *C* und γ für SVMs, Anzahl der Subzentren für MDA und *k* für *k*NN.

Das Ziel des ersten Punktes der obengenannten Fragestellung (Punkt a.) ist die Selektion der optimalen „Tuning“-Parameter. Dabei wird aus einer Vielzahl an möglichen Modellen -vertreten durch die verschiedenen Parametereinstellungen- jeweils eines ausgewählt. Zu diesem Punkt zählt auch die im Vorfeld einer Klassifikation häufig durchgeführte Variablenselektion, bei der aus vielen Variablen ein geeignetes Variablensubset ausgewählt wird. Die Auswahl der „Tuning“-Parameter bzw. eines Variablensubsets wird als Modellselektion bezeichnet.

Der zweite Punkt (Punkt b.) beschäftigt sich mit der Güte des Modells nach stattgefundener Modellselektion. Gemeinsam ist beiden Punkten, dass dabei überprüft wird, wie gut das Modell sogenannte Testdaten (vom Trainingsdatensatz unabhängige Daten) vorhersagen kann. Da nicht immer Testdaten zur Verfügung stehen, muss der vorhandene Datensatz in Trainings- und Testdaten aufgeteilt werden. Auf Basis der so erhaltenen Trainingsdaten wird ein Klassifikationsmodell erstellt, das anschließend zur Vorhersage der Testdaten dient. Als Gütemaß für das Modell wird in der Regel die Wiedererkennungsrate angegeben, die dem Prozentsatz der in der Klassifikation richtig zugeordneten Testdaten entspricht. Findet die Aufteilung in Trainings- und Testdatensatz nur einmal statt, spricht man von der „Hold-out“-Methode. Diese weist einige Nachteile auf [92,93]. Wird beispielsweise ein großer Anteil der Daten als Testdatensatz zur Seite gelegt, basiert das Modell nur auf wenigen Trainingsdaten, was häufig zu einem instabilen Modell und einer schlechten Vorhersageleistung führt. Dies gilt besonders dann, wenn insgesamt nur wenige Daten zur Verfügung stehen. Außerdem ist eine genaue Einschätzung der Vorhersagegenauigkeit bezüglich neuer Daten häufig nicht möglich, da das Modell nicht auf dem gesamten Datensatz sondern auf dem deutlich reduzierten Trainingsdatensatz basiert. Auch bei der Verwendung von wenigen Testdaten in der „Hold-out“-Methode ist die Vorhersage möglicherweise nicht repräsentativ. Das Modell ist zwar stabil, da es auf einem großen Trainingsdatensatz basiert, aber die Varianz der Wiedererkennungsrate bei Verwendung verschiedener kleiner Testdaten ist meistens so hoch, dass die geschätzte Vorhersagegüte keine zuverlässige Aussage erlaubt. Bei einer

„unglücklichen“ Wahl des Testsets, kann die Klassifikationsrate also stark von dem tatsächlich zu erwartenden Wert abweichen. Um diesem Problem entgegenzuwirken, wird das Verfahren des wiederholten Stichprobenziehens (engl. Resampling) angewendet. Die Ergebnisse der Stichproben werden anschließend gemittelt. Zieht man Stichproben ohne Zurücklegen, dann spricht man von Kreuzvalidierung (engl. Cross-Validation, CV) [94,95]. Erfolgt die Stichprobenziehung unter Zurücklegen, dann handelt es sich um die verwandte Technik des „Bootstrappings“ [96,97] (siehe Kapitel 3.4.4.2). In beiden Fällen werden diejenigen Objekte, die nicht Teil der gezogenen Stichprobe sind, als Testdaten verwendet.

Zusätzlich zu den „Resampling“-Techniken wird in dieser Arbeit ein doppeltes Validierungsschema vorgestellt, was folgenden Hintergrund hat: Sollen für ein Klassifikationsproblem sowohl „Tuning“-Parameter festgelegt (siehe Punkt a.) als auch die Güte des ausgewählten Modells abgeschätzt werden (siehe Punkt b.), ist dies normalerweise nicht in einem Kreuzvalidierungsschritt möglich; denn „Tuning“-Parameter, die bei einer einfachen Kreuzvalidierung das beste Ergebnis zeigen, liefern nicht notwendigerweise auch bei der Vorhersage neuer, unbekannter Testobjekte eine gute Klassifikationsrate. Dies lässt sich durch das Auftreten von Zufallskorrelationen erklären, die besonders bei wenigen Objekten und einer großen Anzahl zur Auswahl stehender Modelle beobachtet werden [59,98]. Zufallskorrelationen führen dazu, dass die „Tuning“-Paramter bzw. das Variablensubset, das mittels Kreuzvalidierung ausgewählt wurde, ausschließlich für den bei der Modellselektion zugrundeliegenden Datensatz eine gute Klassifikationsrate zeigt und nicht generell für das Klassifikationsproblem die beste Wahl sein muß. Bei der Vorhersage neuer, unbekannter Daten bewährt sich das gewählte Modell folglich nicht mehr, was dem Problem des „Overfittings“ entspricht (siehe Kapitel 2.3.3.2). Um „Overfitting“ durch Zufallskorrelationen aufzudecken, sollte neben der Kreuzvalidierung zur Modellselektion, die als interne Validierung bezeichnet wird, zusätzlich die Vorhersage eines oder mehrerer unbekannter Testsets, die nicht an der Modellselektion beteiligt sind, durchgeführt werden. Dieser Schritt wird als externe Validierung bezeichnet und dient dazu, die „wahre“ Vorhersagegüte des Modells abzuschätzen. Der Grad des „Overfittings“ bei der Modellselektion (engl. Model Selection Bias) kann über die Differenz der Wiedererkennungsrate zwischen interner und externer Validierung („Bias“) abgeschätzt werden. Durch das in Kapitel 2.3.4.1.3 beschriebene doppelte Validierungsschema [99,100]

ist eine unabhängige Durchführung von Modellselektion und Abschätzung der Modellgüte möglich.

2.3.4.1.1 Kreuzvalidierung

Bei der Kreuzvalidierung (engl. Cross-Validation: CV) [94,95] werden die Daten mehrmals in zwei sich gegenseitig ausschließende Mengen aufgeteilt: In einen Konstruktions- oder Trainingsdatensatz und einen Validier- oder Testdatensatz. Unter Verwendung des Trainingsdatensatzes wird ein Klassifikationsmodell erstellt, das zur Vorhersage der Testdaten dient. Die Ergebnisse der Vorhersage werden dann mit den tatsächlichen Werten verglichen. Daraus wird die Wiedererkennungsrate, die den Prozentsatz der richtig zugeteilten Spektren angibt, berechnet. Nach mehrmaliger Wiederholung mit verschiedenen Untermengen wird die endgültige Klassifikationsrate (mittlerer Prozentsatz der richtig zugeteilten Objekte über alle Testsets) bestimmt. Es gibt unterschiedliche Methoden bezüglich der Aufteilung der Daten in Test- und Trainingsdaten.

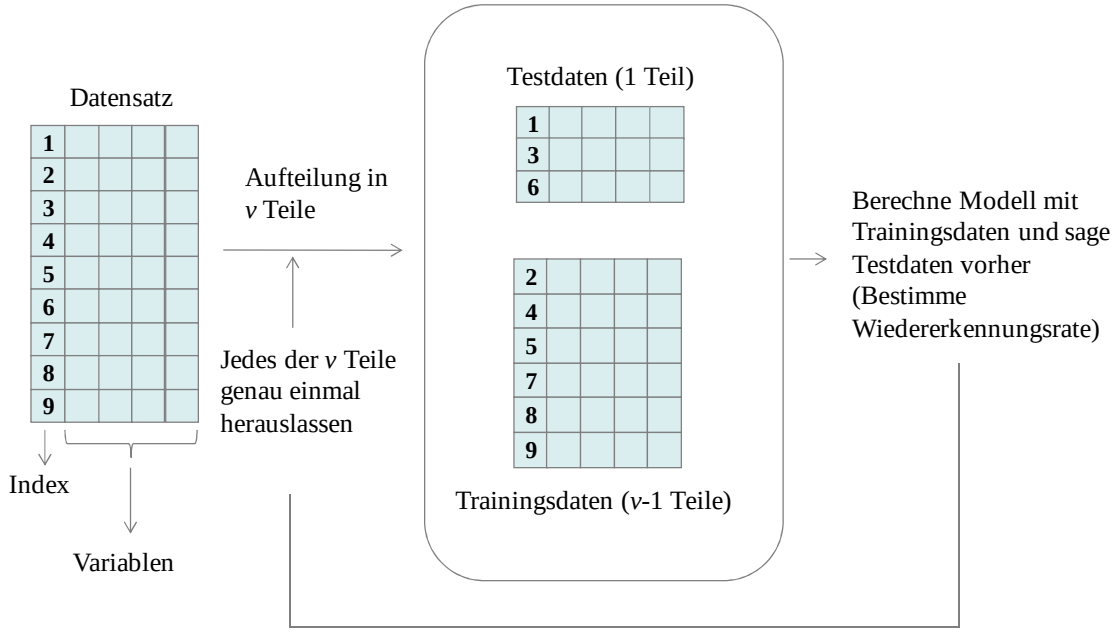


Abbildung 2.18: ν-fache Kreuzvalidierung

Bei der sogenannten ν-fachen Kreuzvalidierung (ν-CV) wird der Datensatz zufällig in ν möglichst gleich große Gruppen unterteilt. Mit ν - 1 Teilen (Trainingsdaten) wird das Modell erstellt und anschließend der verbleibende ν-te Teil (Testdaten) vorhergesagt. Durch ν-malige Wiederholung dieser Prozedur dient jeder der ν Teile genau einmal als Testset (siehe Abbildung 2.8). Ein Spezialfall der ν-fachen Kreuzvalidierung ist die „Leave-One-Out“ Kreuzvalidierung (LOO-CV) [101,102], bei der ν der Anzahl der Objekte n im gesamten Datensatz entspricht. Auf diese Weise wird als Testset immer genau ein Objekt herausgelassen. Ein weiteres beliebtes Kreuzvalidierungsschema ist die „Leave-Multiple-Out“-Kreuzvalidierung (LMO-CV). Dabei werden mehrmals hintereinander eine Anzahl von $d > 1$ Objekten zufällig aus dem Datensatz gezogen und als Testset verwendet. Im Gegensatz zur ν-fachen Kreuzvalidierung landen dabei manche Objekte mehrmals im Testset, manche wiederum gar nicht.

Die vorgestellten Kreuzvalidierungsmethoden sind für die Anwendung in der Modellselektion (siehe Punkt a. in Kapitel 2.3.4.1) sowie für die Schätzung des „wahren“ Vorhersagefehlers eines Modells (Punkt b. in Kapitel 2.3.4.1) nicht gleichermaßen geeignet. Im Folgenden werden die Vor- und Nachteile der Kreuzvalidierungsmethoden für die verschiedenen Anwendungsgebiete diskutiert. Dabei werden nicht nur Studien zur Klassifikation sondern auch zur Regression zitiert, da angenommen wird, dass die Ergebnisse in diesem Kontext auf die Klassifikation übertragbar sind. Anschließend wird für das vorliegende Klassifikationsproblem ein geeignetes Validierungsschema gewählt.

Zunächst wird die Kreuzvalidierung in der Modellselektion (siehe Fragestellung a. in Kapitel 2.3.4.1) betrachtet. In mehreren Studien wurde am Beispiel der Variablenselektion in der Regression (MLR, PCR, PLS) belegt, dass LOO-CV eine starke Tendenz zum „Overfitting“ (siehe Kapitel 2.3.3.2) aufweist [103-106]. So konnte gezeigt werden, dass mittels LOO-CV häufig zu viele Variablen ausgewählt werden. Durch die Hinzunahme unwichtiger Variablen geht also zusätzlich Rauschen (irrelevante Variablen) in die Modellbildung ein. Außerdem wird die Vorhersagegüte der gewählten Modelle überoptimistisch eingeschätzt, so dass bei der Vorhersage unbekannter Testobjekte die Fehlerrate wesentlich höher ist, als während der Kreuzvalidierung zur Variablenselektion geschätzt wurde. Dies tritt v. a. dann auf, wenn aus einer Vielzahl an Modellen ausgewählt werden muss und nur wenige Objekte vorhanden sind. Kohavi beschreibt dieses Phänomen auch in der Klassifikation und empfiehlt statt einer

LOO-CV eine 10-fache Kreuzvalidierung für die Variablenselektion [107]. Shao konnte die Eigenschaften der LOO-CV in der Modellselektion für die multivariate lineare Regression (MLR) mathematisch durch deren asymptotische Inkonsistenz begründen und zeigte, dass durch die Verwendung einer LMO-CV der Hang zur Überoptimierung deutlich reduziert wird [108].

Die Überlegenheit von LMO-CV über LOO-CV und v-facher Kreuzvalidierung in der Modellselektion wurde durch extensive Simulationsstudien von Baumann für PCR und PLS-Regression belegt [105,106]. Baumann konnte dabei einige wichtige Faustregeln für die Wahl der zur Bildung des Modells herausgelassenen Objekte ableiten. So soll die Größe des Testdatensatzes etwa 40%-60% des Trainingsdatensatzes betragen [105,106].

Auch die Schätzung des Vorhersagefehlers nach stattgefundener Modellselektion (siehe Fragestellung b. in Kapitel 2.3.4.1) hängt stark von dem verwendeten Kreuzvalidierungsschema ab [97,109,110]. Bei dem Herauslassen von *d* Objekten als Testset basiert die Schätzung der Modellgüte auf einem Trainingsdatensatz der Größe *n* - *d* [111]. Bei der Vorhersage neuer, unbekannter Testdaten wird jedoch der gesamte Datensatz als Trainingsdatensatz zugrundegelegt. Das Trainingsset enthält dann *n* Objekte. Je kleiner *d* ist, desto mehr wird die Schätzung der Wiedererkennungsrate an die tatsächlich zu erwartende Wiedererkennungsrate bei der Vorhersage unbekannter Testobjekte angenähert. Ermöglicht eine Kreuzvalidierung die Schätzung der „wahren“ Vorhersagegüte, spricht man von einer Validierung ohne „Bias“ (engl. Unbiased Validation). Am kleinsten wird *d* (und somit der „Bias“) bei der LOO-CV, die deshalb zur Abschätzung der Modellgüte sehr gut geeignet ist [109,110,112]. Eine Ausnahme dafür stellen nicht-stabile Klassifikatoren wie z. B. CART (Classification and Regression Trees) [113] dar, für die LOO-CV keine gute Abschätzung der Modellgüte erlaubt. Ein entscheidender Nachteil der LOO-CV ist ihre lange Rechenzeit, die sich vor allem bei der Analyse großer Datensätze bemerkbar macht.

In dieser Arbeit wird die Kreuzvalidierung sowohl für die Modellselektion als auch für die Abschätzung der Güte der gewählten Modelle herangezogen. So werden mittels Kreuzvalidierung verschiedene „Tuning“-Parameter (Anzahl der latenten Variablen für PLS-DA, Anzahl der PCs für LDA, MDA, QDA und *k*NN, C und γ für SVMs, Anzahl der Subzentren für MDA und k for *k*NN) für die Klassifikation ausgewählt. Es findet jedoch keine Variablenselektion statt, bei der aus einer Vielzahl an Variablen ein geeignetes Subset

ausgewählt werden muss, wie es in den genannten Studien der Fall ist. Stattdessen sind die hier zur Auswahl stehenden Parametereinstellungen sehr begrenzt und der zur Verfügung stehende Datensatz ist groß (3642 Objekte). Deshalb wird angenommen, dass das Risiko von Zufallskorrelationen und somit die Gefahr des „Overfittings“ durch Modellselektion gering ist. Es ist also zu erwarten, dass die Anwendung einer LMO-CV, die bei der Modellselektion prinzipiell die geeignete Wahl ist, in diesem Fall keine Vorteile gegenüber einer LOO-CV oder einer v-fachen Kreuzvalidierung bringt. Aus diesem Grund wird in dieser Arbeit bezüglich der Validierungsart ein Kompromiß getroffen. Da hier das Risiko des „Overfittings“ durch Modellselektion aus den genannten Gründen als gering eingestuft wird, wird für den Vergleich der verschiedenen Klassifikationsmodelle das Schema der internen und externen Validierung zunächst umgangen. Sowohl für die Wahl der „Tuning“-Parameter als auch für die Einschätzung der Modellgüte wird eine 50-fache Kreuzvalidierung verwendet. Dabei werden bei jedem Validierungsschritt nur 2% der Daten herausgelassen, was einem kleinen *d* entspricht. Deshalb verhält sich die 50-fache Kreuzvalidierung bezüglich der Abschätzung der Modellgüte ähnlich wie eine LOO-CV, sie erfordert jedoch deutlich weniger Rechenzeit, da statt 3642 Validierungsläufe (für 3642 Objekte im Datensatz) nur 50 Läufe gerechnet werden müssen [114].
Um abschließend die Modellgüte der gewählten Techniken korrekt einschätzen zu können, wird für diejenigen Methoden, die die besten Klassifikationsergebnisse bei der 50-fachen Kreuzvalidierung zeigten, das geforderte Schema aus interner und externer Validierung durchgeführt. Dafür wird eine doppelte Kreuzvalidierung herangezogen, die in Kapitel 2.3.4.1.3 beschrieben ist.

2.3.4.1.2 „Bootstrapping“

„Bootstrapping“ wurde von Bradley Efron auf der Grundlage des „Jackknife“-Verfahrens entwickelt [96,97,115]. Beide Methoden zählen zu den „Resampling“-Verfahren. „Bootstrapping“ wird vor allem dann verwendet, wenn die Wahrscheinlichkeitsverteilung einer Stichprobenfunktion *f*(*x*) nicht mit vertretbarem Aufwand bestimmt werden kann. Um auch in diesen Situationen Verteilungsparameter wie Vertrauensintervalle angeben zu können, wird auf der Grundlage der vorhandenen Daten eine große Anzahl von Pseudo-Zufalls-Datensätzen („Bootstrap“-Stichproben) erzeugt. Diese Pseudo-Datensätze werden

dann verwendet, um die Verteilung der Stichprobenfunktion, insbesondere deren Streuungsparameter, zu schätzen. Die dem „Bootstrapping“ zugrundeliegende Grundannahme besteht darin, dass sich über Verteilungsparameter der einzelnen „Bootstrap“-Stichproben **X*** die Verteilungsparameter der unbekannten Grundgesamtheit schätzen lassen. Beim „Bootstrap“-Verfahren werden die „Bootstrap“-Stichproben **X*** aus dem ursprünglichen Datensatz **X** mit Zurücklegen gezogen. Der Umfang der zufällig gezogenen Stichproben **X*** entspricht dem des zugrundeliegenden Datensatzes **X**. Ziehen mit Zurücklegen bewirkt, dass mit hoher Wahrscheinlichkeit nicht alle Elemente von **X** in **X*** auftreten, dafür aber einige Elemente mehrfach. Aus der Population von Verteilungsparametern der „Bootstrap“-Stichproben lassen sich dann die gewünschten Verteilungsparameter von $f(x)$ herleiten. In dieser Arbeit wird „Bootstrapping“ als Alternative zur Kreuzvalidierung verwendet. Beim Ziehen einer „Bootstrap“-Stichprobe aus dem Datensatz ist die Wahrscheinlichkeit, dass ein Objekt in die Stichprobe gelangt ca. 63%, während die Wahrscheinlichkeit, dass ein Objekt nicht gezogen wird, bei ca. 37% liegt [110]. Die „Bootstrap“-Stichprobe wird als Trainingsdatensatz verwendet. Die Objekte die nicht in der „Bootstrap“-Stichprobe enthalten sind, entsprechen dem Testset. So werden im Vergleich zur 50-fachen Kreuzvalidierung pro Durchlauf wesentlich mehr Daten als Testdaten entnommen (ca. 37% statt 2%). Die einzelnen Trainingsdatensätze basieren beim „Bootstrapping“ zum einen auf weniger Objekten und zum anderen sind deren Zusammensetzungen diverser. Da manche Objekte nicht enthalten sind, manche Objekte dafür mehrfach, kommt es zu einer Verzerrung der einzelnen Trainingsdatensätze gegenüber dem Originaldatensatz. Dies wird in dieser Arbeit genutzt, um auf Robustheit der gewählten Methoden zu prüfen, für den Fall, dass weniger Objekte in den Trainingsdatensatz eingehen. Es wird außerdem das Verhalten der einzelnen Methoden bei der Modellselektion beobachtet, wenn diese auf diversen Datensätzen (im doppelten Validierungsschema, siehe Kapitel 2.3.4.1.3) basiert. Dabei wird „Bootstrapping“ in der äußeren Schleife des doppelten Validierungsschemas verwendet.

2.3.4.1.3 Doppelte Validierung [99,100]

Nur bei der Validierung anhand von Daten, die nicht in die Modellselektion eingeflossen sind, kann sichergestellt werden, dass sich das gewählte Modell auch für die Vorhersage

neuer Daten eignet [59,98]. Deshalb wird neben einer Kreuzvalidierung zur Modellselektion (interne Validierung) zusätzlich eine externe Testdatenvorhersage durchgeführt, bei der die Wiedererkennungsrate unabhängig von dem Modellselektionsprozess bestimmt wird (nachdem alle „Tuning"-Parameter festgelegt sind). In dieser Arbeit wird erwartet, dass der „Bias" der Modellselektion (Differenz zwischen interner und externer kreuzvalidierter Wiedererkennungsrate) gering ist; denn der Datensatz ist groß und die Auswahl der Parametereinstellungen für die Modellselektion ist gering. Für die zwei Klassifikationmethoden mit der besten Wiedererkennungsrate bei der einfachen Kreuzvalidierung (paarweise MDA und SVM) werden die geforderten zwei Validierungsschleifen durchgeführt. Dadurch kann das Ausmaß des „Overfittings" bei Verwendung von nur einer einfachen Kreuzvalidierung abgeschätzt werden (siehe Abbildung 2.19).

Die Durchführung der zwei Kreuzvalidierungsschleifen findet folgendermaßen statt: Die Trainingsdaten die in der äußeren Schleife gebildet werden, werden einer (internen) 50-fachen Kreuzvalidierung unterzogen, um die optimalen „Tuning"-Parameter festzulegen. Die so bestimmten „Tuning"-Parameter werden wiederum verwendet, um das externe Testset vorherzusagen. Da die externe Schleife hier aus einer 50-fachen Kreuzvalidierung besteht, findet dieser Vorgang 50 Mal statt. Die besten „Tuning"-Parameter der inneren Schleife müssen nicht für jeden der 50 externen Durchläufe dieselben sein. Deshalb werden variierende Parameter für die 50 externen Testset-Vorhersagen verwendet. Die durchschnittliche Wiedererkennungsrate sowie die Standardabweichung der externen Testdatenvorhersagen werden dokumentiert. Beim Vergleich dieser Ergebnisse mit den Werten aus der einfachen 50-fachen Kreuzvalidierung mit festgelegten Parametern können Rückschlüsse über den Einfluss der Datensatzzusammensetzung auf die Auswahl der „Tuning"-Parameter und die Klassifikationsrate gezogen werden.

Um außerdem die Robustheit der Klassifikationsmethoden gegenüber der Verwendung kleinerer und diverserer Trainingsdatensätze abzuschätzen, wird zusätzlich die doppelte Kreuzvalidierung mit einem „Bootstrapping" in der äußeren Schleife und einer 50-fachen Kreuzvalidierung in der inneren Schleife durchgeführt (siehe Kapitel 3.4.4.2).

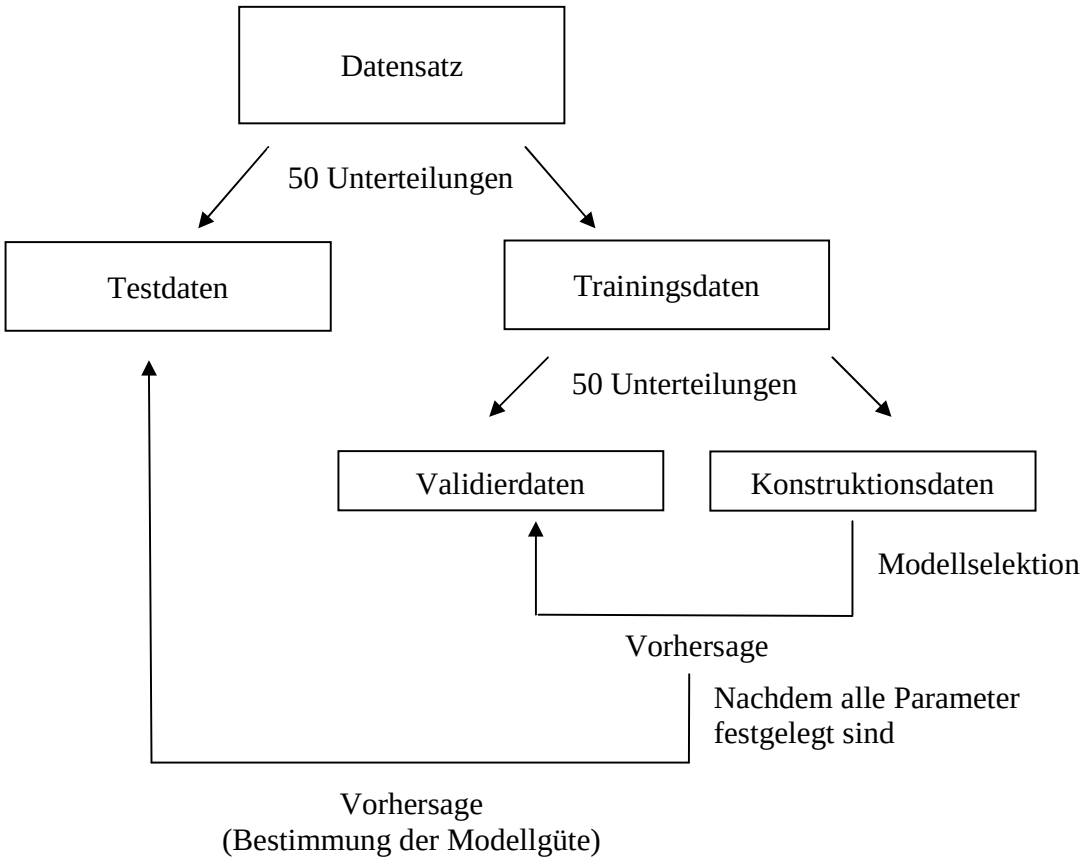


Abbildung 2.19: Validierungsschema mit 2 Schleifen der 50-fachen Kreuzvalidierung

2.3.4.2 Kreuzvalidierte Varianzanalyse (CVANOVA)

Im Rahmen dieser Arbeit werden verschiedene Modelle bezüglich ihrer Leistungsfähigkeit für die Klassifikation von Mikroorganismen beurteilt und miteinander verglichen. Dabei stellt sich nicht nur die Frage, welche Methode bzw. Methodenkombination in der Kreuzvalidierung die beste Klassifikationsrate erzielt, sondern auch, wie die Unterschiede zwischen den einzelnen Ergebnissen zu bewerten sind. Zu prüfen ist also, ob die Unterschiede zwischen den Klassifikationsraten signifikant sind oder ob sie so klein sind,

dass sie wahrscheinlich durch zufälliges Rauschen entstanden sind. Derartige Fragestellungen können mit Hilfe einer Varianzanalyse (engl. Analysis of Variance: ANOVA) [116] untersucht werden. Man unterscheidet verschiedene Formen der auf der F-Statistik basierenden Varianzanalyse. Beim Vorliegen einer Zielvariablen (abhängige Variable), spricht man von einer univariaten Varianzanalyse (ANOVA). Dagegen wird eine Varianzanalyse mit mehreren Zielvariablen als multivariaten Varianzanalyse (MANOVA) bezeichnet. In dieser Arbeit stellt die aus der Kreuzvalidierung erhaltene Klassifikationsrate die Zielvariable dar. Es handelt sich also um eine univariate Varianzanalyse (ANOVA). Auch die Anzahl an Einflussfaktoren (unabhängige Variable) kann variieren. Wird die Auswirkung von nur einer unabhängigen Variable (Einflussfaktor) auf eine abhängige Variable (Zielvariable) untersucht, spricht man von einer einfaktoriellen Varianzanalyse (engl. One-Way ANOVA). Bei zwei oder mehr Einflussfaktoren spricht man von einer zwei- bzw. mehrfaktoriellen Varianzanalyse (engl. Two-Way ANOVA, Multi-Way ANOVA). In dieser Arbeit kommt sowohl die ein- als auch die zweifaktorielle ANOVA zum Einsatz. So kann beispielsweise mittels zweifaktorieller ANOVA simultan der Einfluss verschiedener Vorbehandlungsmethoden und Klassifikationsmethoden auf die Klassifikationsrate evaluiert werden.

Durch den Vergleich der Gruppenmittelwerte und der Varianz der Zielvariablen, können Aussagen über die Signifikanz und Stärke der Einflussfaktoren getroffen werden. Die einfaktorielle ANOVA ist eng verwandt mit dem einfachen t-Test, der darauf beschränkt ist, Mittelwerte zweier Gruppen auf Unterschiedlichkeit zu testen. So führen t-Test und einfaktorielle ANOVA im Zwei-Gruppen-Fall zum gleichen Ergebnis. Allerdings ist es mit Hilfe der einfaktoriellen ANOVA möglich, mehr als zwei Gruppenmittelwerte simultan zu vergleichen.

Für den speziellen Fall des Vergleichs von kreuzvalidierten Vorhersageergebnissen führten Indahl und Naes die sogenannte "Cross-Validated ANOVA" (CVANOVA) ein [117]. Dabei analysierten sie kreuzvalidierte Regressionsmodelle durch eine zweifaktorielle ANOVA. In der vorliegenden Arbeit wird CVANOVA für die Auswertung von kreuzvalidierten Klassifikationsergebnissen verwendet.

2.3.4.2.1 Voraussetzungen

Für die Durchführung einer ein- oder zweifaktoriellen ANOVA bzw. CVANOVA werden folgende Anforderungen an die Daten gestellt:

- Normalverteilung der Stichproben
- Varianzhomogenität (Homoskedaszidität) der Stichproben
- Unabhängigkeit der Stichproben

Weichen die untersuchten Daten stark von diesen Anforderungen ab, sind die erhaltenen Signifikanz-Aussagen nicht zuverlässig. Da experimentelle Daten diese Kriterien nie perfekt erfüllen, haben sich zahlreiche Studien damit beschäftigt, in welchem Ausmaß Abweichungen von diesen Annahmen die Ergebnisse beeinflussen [118,119]. Während einige Autoren warnen, dass auch leichte Abweichungen von den Anforderungen gravierende Konsequenzen bezüglich der Gültigkeit der Ergebnisse haben können [120], kommen zahlreiche andere Studien zu dem Schluss, dass sich die einfaktorielle ANOVA sowie der verwandte t-Test robust gegenüber nicht zu groben Abweichungen verhalten, wenn bestimmte Regeln eingehalten werden. So werden Abweichungen vor allem dann toleriert, wenn viele Daten für die Berechnung zur Verfügung stehen und wenn der Stichprobenumfang pro Gruppe gleich groß ist [119,121-124]. In der hier verwendeten CVANOVA enthält jede zu vergleichende Gruppe 50 Vorhersagewerte, da eine 50-fache Kreuzvalidierung durchgeführt wird. Es handelt sich also um eine relativ große Datenmenge. Zudem ist der Probenumfang jeder Gruppe gleich, so dass eine gute Robustheit gegenüber leichten Abweichungen gegeben sein sollte. Um festzustellen ob die Kriterien der Normalverteilung und Varianzhomogenität annähernd erfüllt sind, werden in der Literatur verschiedene Tests vorgeschlagen. In der Regel handelt es sich dabei um Betrachten von Residuenplots, Normal-Quantil-Quantil-Plots (Normal-Q-Q-Plots) [125] oder um Signifikanztests, wobei die Anwendung von Signifikanztests zum Teil kritisch beurteilt wird [126-128]. Um in dieser Arbeit zu untersuchen, ob grobe Abweichungen von der Normalverteilung bestehen, wurden Normal-Q-Q-Plots betrachtet. Daneben wurde Bartlett´s Test zur Prüfung auf Varianzhomogenität durchgeführt. Zu den theoretischen Grundlagen dieser beiden Verfahren siehe [125,129]. Stellt man bei der Durchführung der

Signifikanztests grobe Abweichungen von Normalverteilung und Varianzhomogenität fest, sollten die Daten vor der Durchführung einer CVANOVA transformiert werden (z. B. Logarithmus-Transformation, Wurzeltransformation und Arcus-Sinus-Wurzeltransformation) [130]. Alternativ dazu schlugen Van der Voet [131] und Thomas [132] für den Vergleich kreuzvalidierter Vorhersageergebnisse die Verwendung nicht-parametrischer Tests vor. Einen umfassenden Vergleich parametrischer und nicht-parametrischer Signifikanztests zur Beurteilung kreuzvalidierter Vorhersageergebnisse für verschiedene Datensätze veröffentlichten Cederkvist et al. [133]. Cederkvist stellte fest, dass CVANOVA für den Vergleich von Regressionsmodellen in der Regel zuverlässige Aussagen liefert und dass die Unterschiede der Ergebnisse zwischen parametrischen und nicht-parametrischen Signifikanztests häufig gering sind.

In dieser Arbeit wird aufgrund von leichten Abweichungen von Normalverteilung sowie Varianzhomogenität (siehe Kapitel 3.4.1.1.1) parallel zur einfaktoriellen CVANOVA ein nicht-parametrischer Signifikanztest (Kruskal-Wallis, siehe Kapitel 2.3.4.3) durchgeführt. Dadurch können Fehler, die bei einer CVANOVA aufgrund von Abweichungen von den Annahmen entstehen können, aufgedeckt werden.

Die Anforderung der Unabhängigkeit der Stichproben hängt in dem Fall der CVANOVA (und auch beim Kruskal-Wallis Test, siehe Kapitel 2.3.4.3) von der Art der Kreuzvalidierung ab. Bei der 50-fachen Kreuzvalidierung sind die Testdaten voneinander unabhängig, da bei jeder der 50 Testdatenvorhersagen kein Objekt zweimal im Testset auftaucht. Es überlappen allerdings die Objekte in den Trainingssets. Aus diesem Grund sind die 50 Vorhersagewerte aus der Kreuzvalidierung nicht unabhängig voneinander. Die Auswirkung der beschriebenen Abhängigkeit auf die Zuverlässigkeit eines kreuzvalidierten t-Tests, wird in einer Studie von Dietterich beschrieben [134]. Da man den t-Test als Spezialfall der ANOVA ansehen kann, werden die Ergebnisse dieser Studie auch für eine ANOVA als gültig angesehen. Dietterich zeigte, dass bei einer ν-fachen Kreuzvalidierung der t-Test leicht erhöhte Irrtumswahrscheinlichkeiten α liefert. Dagegen wurden bei der Anwendung der LMO-CV starke Abweichungen vom α-Fehler erhalten, weshalb die LMO-CV für diese Auswertung nicht geeignet ist (hohe Abhängigkeit sowohl der Test- als auch der Trainingsdatensplits). Der Vergleich kreuzvalidierter Vorhersageergebnisse mittels CVANOVA dient in dieser Arbeit als Hilfe zur Zusammenfassung und Beurteilung der zahlreichen Ergebnisse. Leicht

höhere Signifikanzwerte als die angegebenen (Abhängigkeit der Trainingssets in der 50-fachen Kreuzvalidierung) können in Kauf genommen werden.

2.3.4.2.2 Einfaktorielle CVANOVA

Ein Beispiel für eine einfache kreuzvalidierte ANOVA ist der Vergleich verschiedener Klassifikationsmethoden (z. B. LDA, QDA und *k*NN) für ein bestimmtes Klassifikationsproblem. Bei der Durchführung einer 50-fachen Kreuzvalidierung sind für jede Klassifikationsmethode 50 Wiedererkennungsraten (jeweils für einen der 50 Testdatensätze der Kreuzvalidierung) vorhanden. Diese aus der Kreuzvalidierung stammenden Teilwiedererkennungsraten werden im Folgenden Einzelvorhersagewerte genannt. Der Mittelwert der 50 Einzelvorhersagewerte entspricht der Klassifikationsrate. Folgende Fragen sollen beantwortet werden:

- Besteht ein signifikanter Unterschied zwischen den Klassifikationsraten?
- Wenn ja, welche Werte unterscheiden sich voneinander und welche nicht?

Die einfaktorielle ANOVA geht von einer zu testenden Grundannahme, der sogenannten Nullhypothese H_0, aus. Die Nullhypothese wird dahingehend überprüft, ob sie widerlegt werden kann. H_0 lautet hier: Zwischen den Klassifikationsraten besteht kein Unterschied. Bei Verwerfen der Nullhypothese, tritt die Alternativhypothese H_1 in Kraft, welche besagt: Die Klassifikationsraten unterscheiden sich. Bei einem Signifikanzlevel α von 0.05 ist die Wahrscheinlichkeit, die Nullhypothese fälschlicherweise zu verwerfen, 5% (Irrtumswahrscheinlichkeit). Zur Lösung der genannten Fragestellungen wird die Gesamtvarianz in verschiedene Varianzteile zerlegt.

Ist *k* die Anzahl der Klassifikationsmethoden (im Folgenden Gruppen genannt), *n* die Gesamtzahl der Einzelvorhersagewerte (hier 3x50, da die kreuzvalidierten Ergebnisse von 3 Klassifikationsmethoden verglichen werden), n_i die Anzahl der Einzelvorhersagewerte pro Gruppe (hier 50) und x_{io} der *o*-te Einzelvorhersagewert der *i*-ten Klassifikationsmethode bzw. Gruppe so beträgt die Gesamtvarianz:

$$\text{Gesamtvarianz} = \frac{1}{n-1} \cdot \sum_{i=1}^{k} \sum_{o=1}^{n_i} (x_{io} - \bar{\bar{x}})^2 \tag{2.62}$$

Dabei steht $\bar{\bar{x}}$ für den Mittelwert über alle Einzelvorhersagewerte (sog. „Grand Mean"). Nun werden die Varianzen zwischen und innerhalb der einzelnen Gruppen verglichen. Je größer dieses Varianzverhältnis ist, desto größer ist die Wahrscheinlichkeit, dass die unabhängige Variable (hier Klassifikationsmethode) einen Einfluss auf die abhängige Variable (hier Klassifikationsergebnis) hat.

Die Varianz ist proportional zur Summe der Abweichungsquadrate – kurz SS für „Sum of Squares". Die gesamten Summen der Abweichungsquadrate (SS_{TOTAL}) werden in die Quadratsummen innerhalb (SS_{WITHIN}) und die Quadratsummen zwischen den Gruppen ($SS_{BETWEEN}$) zerlegt (Gl. (2.63)(2.64)).

$$SS_{TOTAL} = SS_{WITHIN} + SS_{BETWEEN} \tag{2.63}$$

$$\sum_{i=1}^{k} \sum_{o=1}^{n_i} (x_{io} - \bar{\bar{x}})^2 = \sum_{i=1}^{k} \sum_{o=1}^{n_i} (x_{io} - \bar{x}_i)^2 + \sum_{i=1}^{k} (n_i \cdot (\bar{x}_i - \bar{\bar{x}})^2) \tag{2.64}$$

Dabei entspricht $\bar{x}_i$ der Klassifikationsrate der *i*-ten Klassifikationsmethode (Mittelwert über alle Einzelvorhersagewerte der Gruppe *i*). Anschließend werden die Quadratsummen durch ihre zugehörige Anzahl an Freiheitsgraden (*df* für „Degrees of Freedom") geteilt und man erhält die mittleren Abweichungsquadrate (MS für „Mean Squares") (Gl. (2.65)-((2.68)). Diese entsprechen den geschätzten Varianzen zwischen und innerhalb der Gruppen. Sie

werden zueinander ins Verhältnis gesetzt und man erhält die Prüfgröße F (Gl. (2.69)). Je größer der F-Wert ist, desto größer sind die Unterschiede zwischen den Gruppen.

$$df_{BETWEEN}=k\text{-}1 \tag{2.65}$$

$$df_{WITHIN}=n\text{-}k \tag{2.66}$$

$$MS_{BETWEEN}=\frac{SS_{BETWEEN}}{df_{BETWEEN}} \tag{2.67}$$

$$MS_{WITHIN}=\frac{SS_{WITHIN}}{df_{WITHIN}} \tag{2.68}$$

$$F=\frac{MS_{BETWEEN}}{MS_{WITHIN}} \tag{2.69}$$

Der empirisch ermittelte F-Wert wird mit dem theoretischen Wert (5%-Niveau) der F-Tabelle verglichen. Überschreitet der empirische den theoretischen Wert, wird die Nullhypothese H_0 verworfen und die Hypothese H_1 angenommen.

2.3.4.2.3 Zweifaktorielle CVANOVA

Die zweifaktorielle ANOVA prüft als Erweiterung der einfaktoriellen ANOVA die Auswirkung von zwei unabhängigen Variablen (Faktoren A und B) auf eine abhängige Variable. Dabei wird beispielsweise der Einfluss verschiedener Vorbehandlungsmethoden (Faktor A) und Klassifikationsmethoden (Faktor B) auf das Klassifikationsergebnis analysiert. Die Ausprägungen eines Faktors nennt man Faktorstufen. Eine Faktorstufe entspricht einer Vorbehandlungs- bzw. Klassifikationsmethode. Die sich unter den verschiedenen Kombinationen der Faktorstufen ergebenden Gruppen nennt man Zellen. Eine Zelle entspricht der Kombination einer Vorbehandlungsmethode und einer Klassifikationsmethode. Es ist möglich, dass die beiden Faktoren nicht nur direkt auf das beobachtete Merkmal (Klassifikationsergebnis) wirken, sondern dass auch Wechselwirkungen bestehen.

Deshalb überprüft die zweifaktorielle Varianzanalyse drei voneinander unabhängige Nullhypothesen H_0:

- Die unter den Stufen des Faktors A beobachteten Mittelwerte unterscheiden sich nicht.
- Die unter den Stufen des Faktors B beobachteten Mittelwerte unterscheiden sich nicht.
- Zwischen den beiden Faktorstufenkombinationen besteht keine Interaktion.

Zur Berechnung der zweifaktoriellen Varianzanalyse werden die summierten Abweichungsquadrate in vier Teile zerlegt. Die bei der einfaktoriellen ANOVA berechneten Quadratsummen zwischen den Gruppen ($SS_{BETWEEN}$) werden dabei nochmals in drei Quadratsummen (SS_A, SS_B und SS_{AxB}) aufgeteilt. SS_A und SS_B entsprechen den Abweichungsquadraten, die jeweils durch die Faktoren A bzw. B einzeln verursacht wurden. SS_{AxB} beschreibt den Quadratsummen-Anteil der auf die Wechselwirkung zwischen Faktor A und Faktor B zurückgeht. SS_{WITHIN} ist der durch Rauschen entstandene Varianzanteil und ist analog der Variablen SS_{WITHIN} der einfaktoriellen Varianzanalyse.

$$SS_{TOTAL}=SS_A+SS_B+SS_{AxB}+SS_{WITHIN} \tag{2.70}$$

Die Einzelterme aus Gleichung (2.70) setzen sich wie folgt zusammen:

$$SS_{TOTAL}=\sum_{i=1}^{M}\sum_{j=1}^{N}\sum_{o=1}^{n_{ij}}(x_{ijo}-\bar{\bar{x}}) \tag{2.71}$$

$$SS_A=\sum_{i=1}^{M}(n_i\cdot(\bar{x}_i-\bar{\bar{x}})^2) \tag{2.72}$$

$$SS_B=\sum_{j=1}^{N}(n_j\cdot(\bar{x}_j-\bar{\bar{x}})^2) \tag{2.73}$$

$$SS_{AxB}=\sum_{i=1}^{M}\sum_{j=1}^{N}(n_{ij}\cdot(\bar{x}_{ij}-\bar{\bar{x}})^2)-\sum_{i=1}^{M}(n_i\cdot(\bar{x}_i-\bar{\bar{x}})^2)-\sum_{j=1}^{N}(n_j\cdot(\bar{x}_j-\bar{\bar{x}})^2) \quad (2.74)$$

$$SS_{WITHIN}=\sum_{i=1}^{M}\sum_{j=1}^{N}\sum_{o=1}^{n_{ij}}(x_{ijo}-\bar{x}_{ij}) \quad (2.75)$$

Dabei entsprechen *M* und *N* der Anzahl an Stufen für Faktor A und Faktor B (Anzahl der Vorbehandlungs- und Klassifikationsmethoden). n_{ij} steht für die Anzahl aller Werte einer Zelle und $\bar{x}_{ij}$ für den dazugehörigen Mittelwert. n_i und $\bar{x}_i$ entsprechen der Anzahl und dem Mittelwert aller Werte für die Stufe *i* des Faktors A. $\bar{x}_j$ und n_j sind die korrespondierenden Werte für Faktor B. Das Teilen der Quadratsummen durch ihre zugehörige Anzahl an Freiheitsgraden *df* ergibt analog zur einfaktoriellen ANOVA die mittleren Abweichungsquadrate, aus welchen wiederum die empirischen F-Werte berechnet werden können:

$$df_A=M-1 \quad (2.76)$$

$$df_B=N-1 \quad (2.77)$$

$$df_{AxB}=(M-1)\cdot(N-1) \quad (2.78)$$

$$df_{WITHIN}=n-M\cdot N \quad (2.79)$$

$$MS_A=\frac{SS_A}{df_A} \quad (2.80)$$

$$MS_B=\frac{SS_B}{df_B} \quad (2.81)$$

$$MS_{AxB}=\frac{SS_{AxB}}{df_{AxB}} \quad (2.82)$$

$$MS_{WITHIN} = \frac{SS_{WITHIN}}{df_{WITHIN}} \tag{2.83}$$

$$F_A = \frac{MS_A}{MS_{WITHIN}} \tag{2.84}$$

$$F_B = \frac{MS_B}{MS_{WITHIN}} \tag{2.85}$$

$$F_{AxB} = \frac{MS_{AxB}}{MS_{WITHIN}} \tag{2.86}$$

2.3.4.2.4 „Post-Hoc"-Tests

Wurde durch einfaktorielle oder zweifaktorielle ANOVA festgestellt, dass es Abweichungen zwischen den Mittelwerten gibt, kann mit „Post-Hoc"-Tests festgestellt werden, welche Mittelwerte sich unterscheiden [135]. Auf den ersten Blick bietet sich für paarweise Vergleiche der t-Test nach Student an. Korrekter ist es aber einen Test zu verwenden, der auf der Varianzanalyse aufbaut und Zwischenergebnisse dieses Verfahrens benutzt. Solche Tests nennt man „Post-Hoc"-Tests. Durch diese wird im Gegensatz zum t-Test die Kumulierung des α-Fehlers; d.h. das Steigen der Irrtumswahrscheinlichkeit durch multiples Testen, vermieden. In dieser Arbeit wird für diesen Zweck der Tukey-Test verwendet, der auch Tukey's HSD (Honestly Significant Difference) Test genannt wird. Er basiert auf der studentisierten Spannweitenverteilung. Diese ist der t-Verteilung ähnlich, berücksichtigt aber die Anzahl der verglichenen Mittelwerte und kann deshalb auf den paarweisen Vergleich von mehr als zwei Mittelwerten angewendet werden. Je mehr Gruppenmittelwerte verglichen werden, desto größer ist der tabellierte kritische Wert κ, ab welchem zwei Mittelwerte als signifikant unterschiedlich definiert sind.

Zur Berechnung des empirischen Spannweiten-Verhältnisses κ werden die zwei Gruppenmittelwerte $\bar{x}_i$ und $\bar{x}_j$ mit der Wurzel der mittleren Standardabweichung MS_{WITHIN} innerhalb der Gruppen geteilt durch die Stichprobenanzahl pro Gruppe n_i (=n_j) ins Verhältnis gesetzt.

$$\kappa = \frac{\bar{x}_i - \bar{x}_j}{\sqrt{\frac{MS_{WITHIN}}{n_i}}} \tag{2.87}$$

κ wird mit dem tabellierten kritischen Wert $\kappa_{\alpha,k,df_{WITHIN}}$ für die studentisierte Spannweiten-Verteilung verglichen, der von den Faktoren α (Wahrscheinlichkeit eine wahre Nullhypothese zu verwerfen), k (Anzahl der Gruppen) und df_{WITHIN} (Anzahl der Fehlerfreiheitsgrade) abhängt. Wenn κ größer als der tabellierte Wert $\kappa_{\alpha,k,df_{WITHIN}}$ ist, sind die Mittelwerte signifikant unterschiedlich.

2.3.4.3 Kruskal-Wallis-Test

Sind die in Kapitel 2.3.4.2.1 beschriebenen Kriterien für die Durchführung einer ein- bzw. zweifaktoriellen ANOVA (Normalverteilung, Varianzhomogenität) nicht erfüllt und führt eine Transformation der Daten nicht zu einer Verbesserung, sollten stattdessen nicht-parametrische Signifikanztests herangezogen werden, die diese Kriterien nicht voraussetzen [120]. Ein Beispiel für einen nicht-parametrischen Test ist der Kruskal-Wallis-Test [48,136], der auch H-Test genannt wird und eine nicht-parametrische Version der einfaktoriellen ANOVA darstellt. Sind die Daten annähernd normalverteilt, sind die parametrischen Versionen der ANOVA den nicht-parametrischen Tests vorzuziehen, da sie in dem Fall eine größere „Power“ (Fähigkeit vorhandene Unterschiede zu erkennen), aufweisen [137]. Nicht-parametrische Tests verhalten sich häufig konservativer als parametrische, d.h. sie bestätigen eher die Nullhypothese H_0. In dieser Arbeit wurde festgestellt, dass der Großteil der Daten den Anforderungen einer ein- bzw. zweifaktoriellen ANOVA entspricht, während nur einige,

wenige Gruppen leichte Abweichungen von Normalverteilung bzw. Varianzhomogenität zeigen (siehe Kapitel 3.4.1.1.1). Aufgrund der Menge an Daten und aufgrund gleicher Stichprobengrößen, ist davon auszugehen, dass die Tests sich relativ robust gegenüber derartigen Abweichungen verhalten. Um dennoch entstehende Fehler kontrollieren zu können, werden neben den Ergebnissen der einfaktoriellen ANOVA zusätzlich die Ergebnisse des Kruskal-Wallis-Tests gezeigt (siehe Tabelle 3.3). Die Nullhypothese H_0 sowie die Alternativhypothese H_1 des Kruskal-Wallis-Tests entsprechen denen einer ein- bzw. zweifaktoriellen ANOVA. Die Berechnung basiert auf Rangplatzsummen. Dabei werden zunächst die *n* Einzelvorhersagewerte aller zu vergleichenden Gruppen in aufsteigender Reihenfolge geordnet und mit den Rängen 1 bis *n* versehen. Haben einige Beobachtungen den gleichen Wert, wird ihnen der Durchschnitt der entsprechenden Rangzahlen zugeordnet. Zu jeder Rangzahl wird die Gruppe, aus der die Beobachtung stammt, vermerkt. Im Anschluss daran wird die Summe der Ränge für die einzelnen Gruppen gebildet. R_i (i = 1,2,…,*k)* entspricht der Summe der Ränge der *i*-ten Gruppe. Kruskal und Wallis haben folgende Größe als Teststatistik vorgschlagen:

$$\hat{H} = \left[\frac{12}{n \cdot (n+1)}\right] \cdot \left[\sum_{i=1}^{k} \frac{R_i^2}{n_i}\right] - 3 \cdot (n+1) \tag{2.88}$$

$\hat{H}$ ist annähernd χ^2-verteilt mit *k*-1 Freiheitsgraden und entspricht der geschätzten Varianz der Gruppen-Rangsummen R_i für hinreichend große Werte *n* ($n \geq 5$) und *k* ($k \geq 4$). H_0 wird abgelehnt, sobald $\hat{H}$ größer ist, als der tabellierte Wert $\chi^2_{k-1;\alpha}$. Dabei beschreibt α wiederum die Irrtumswahrscheinlichkeit. Ist $\hat{H}$ auf dem gewählten Niveau statistisch signifikant, kann der Test nach Tukey (siehe Kapitel 2.3.4.2.4) angewendet werden, um zu entscheiden, welche Gruppen sich im Einzelnen unterscheiden. Hierzu werden die mittleren Ränge $\overline{R}_i$ der einzelnen Gruppen berechnet, indem durch ihre Stichprobenumfänge n_i dividiert wird: $\overline{R}_i = R_i / n_i$.

Für signifikante Unterschiede zwischen zwei Gruppen *i* und *j* gilt [48]:

$$|\overline{R}_i - \overline{R}_j| > \frac{\kappa_{\alpha,k,\infty}}{\sqrt{2}} \cdot \sqrt{\left[\frac{n \cdot (n+1)}{12}\right] \cdot \left[\frac{1}{n_i} + \frac{1}{n_j}\right]} \tag{2.89}$$

Dabei entspricht $\kappa_{\alpha,k,\infty}$ dem tabellierten Wert für die studentisierte Spannweitenverteilung.

2.3.5 Clusteranalyse

Analog zur Klassifikation werden auch bei der Clusteranalyse Objekte entsprechend ihrer Ähnlichkeiten gruppiert. Im Gegensatz zur Klassifikation ist die Gruppenzugehörigkeit allerdings im Vorfeld nicht bekannt, weswegen die Clusteranalyse zu den unüberwachten Lernmethoden zählt. Es gibt verschiedenste Arten und Einteilungsmöglichkeiten für Cluster-Algorithmen. So unterscheidet man zwischen Partitionierungsverfahren und hierarchischen Verfahren. Bei Partitionierungsverfahren werden *k* Cluster vorgegeben. Die Cluster werden daraufhin so lange optimiert bis sie untereinander möglichst heterogen und innerhalb jedes Clusters möglichst homogen sind. Das prominenteste Beispiel für dieses iterative Verfahren ist der *k*-Means Algorithmus. (siehe Algorithmus 2.3). Die hierarchischen Cluster Verfahren teilt man ein in agglomerative und divisive Methoden. Von einer agglomerativen Methode spricht man, wenn mit *n* (Anzahl aller Datenpunkte) Clustern begonnen wird und schrittweise die ähnlichsten Cluster zusammengefasst werden. Bei der divisiven Methode beginnt man hingegen mit nur einem Cluster und teilt diesen in immer heterogenere Cluster auf. Neben der Einteilung in Partitionierungsverfahren und hierarchische Cluster-Verfahren erfolgt zudem eine Kategorisierung in „harte" und „weiche" Cluster-Algorithmen. Harte Methoden (z. B. *k*-Means) ordnen jeden Datenpunkt genau einem Cluster zu, wohingegen bei weichen Methoden (z. B. Fuzzy-*k*-Means, „Gaussian-Mixture"-Modell: siehe Kapitel 2.3.3.4.4) jedem Datenpunkt bezüglich jedes Clusters eine Wahrscheinlichkeit zugeordnet wird, mit der der Datenpunkt dem jeweiligen Cluster angehört. Eine weitere Möglichkeit des

unüberwachten Lernens bieten die Karten nach Kohonen (engl. „Self Organizing Maps“: SOMs) (siehe Kapitel 2.3.5.2.) Diese ermöglichen es, hochdimensionale Daten unter Erhaltung der Topologie auf einen niederdimensionalen Raum zu projizieren und so die Datenstrukturen im 2- oder 3-Dimensionalen visualisierbar zu machen. In dieser Arbeit wird die Clusteranalyse verwendet, um den Effekt verschiedener Kultivierungsbedingungen auf die Datenstruktur zu veranschaulichen. Dafür wird zum einen mit dem EM-Algorithmus für Gauss´sche Mischmodelle („Gaussian Mixtures“, siehe Kapitel 2.3.3.4.4) geclustert. Dieser Algorithmus wird bereits für die Klassifikation der Bakterien durch MDA verwendet und kann hier zusätzlich als Werkzeug zur Interpretation dienen. Um die daraus gewonnenen Ergebnisse zu veranschaulichen, wird außerdem eine Visualisierung mit Hilfe von Kohonenkarten (SOMs) durchgeführt.

2.3.5.1 „Gaussian Mixtures" und die empirische bedingte Entropie

Wie in Tabelle 3.1 gezeigt, wurden die in dieser Arbeit untersuchten Bakterienstämme auf verschiedenen Wachstumsmedien, mit unterschiedlichen Temperaturen und Wachstumszeiten kultiviert.

Durch die Clusteranalyse soll nun der Effekt der verschiedenen Kultivierungsbedingungen auf die Datenstruktur dargestellt werden. Bei der Verwendung der „Gaussian Mixtures“ für die Clusteranalyse wird ein MDA-Modell (4 Subzentren und 30 PCs) auf dem gesamten Datensatz trainiert. Anschließend wird für die einzelnen Bakterienstämme untersucht, ob die Kultivierungsparameter in den 4 Subzentren der MDA wiedergefunden werden können; d.h. ob eine Übereinstimmung zwischen den Wachstumsbedingungen und der Verteilung der Spektren auf die Subzentren der MDA besteht. Dabei wird ein Wachstumsparameter jeweils als eine Klasse behandelt. Nun wird der Grad der Überlappung zwischen Klassenzugehörigkeit c und der Zuordnung zu den Subzentren r geschätzt. C steht dabei für die Anzahl der Klassen und R für die Anzahl der Subzentren. In der Literatur findet man mehrere Algorithmen, die die Güte von Clustermethoden bei der Gruppierung von Daten mit bekannten Klassen abschätzen. In dieser Arbeit wird ein Ansatz verwendet, der auf einer sogenannten Kontingenztafel basiert. Eine Kontingenztafel ist eine Matrix der Größe CxK, wobei jede Reihe einer Klasse c (hier Kultivierungsparameter) und jede Spalte einem Cluster

k (hier Subzentrum der MDA) entspricht (siehe Tabelle 3.10 und Tabelle 3.11). Um das Ausmaß der Überlappung zwischen Klassen und Clustern zu quantifizieren, wird die sogenannte empirische bedingte Entropie (engl. Empirical Conditional Entropy: ECE) [138] berechnet. ECE ist der geschätzte Wert $\hat{H}(c \mid r)$ für die bedingte Entropie $H(c|k)$ mit unbekanntem $p(c,r)$. $p(c,r)$ bezeichnet die Wahrscheinlichkeit, dass ein Objekt in die Klasse *c* und gleichzeitig in Cluster *r* fällt.

$$H(c \mid r) = -\sum_{c=1}^{C}\sum_{r=1}^{R} p(c,r) \cdot log\big(p(c \mid r)\big) \tag{2.90}$$

$$\text{ECE} = \hat{H}(c \mid \text{r}) = -\sum_{c=1}^{C}\sum_{r=1}^{R} \frac{h(c,r)}{n} \cdot log\left(\frac{h(c,r)}{h(r)}\right) \tag{2.91}$$

Dabei entspricht $h(c,r)$ der Anzahl an Objekten in Klasse *c* und Cluster *r*. Die Variable *n* beschreibt die Größe des Datensatzes $\left(n = \sum_{c=1}^{C}\sum_{r=1}^{R} h(c,r)\right)$ und die Funktion $h(r)$ entspricht der Summe aller Objekte in Cluster *r* $\left(h(r) = \sum_{c=1}^{C} h(c,r)\right)$. ECE kann Werte zwischen 0 und 1 annehmen. Je kleiner der Wert ist, desto besser ist die Übereinstimmung zwischen Klassen und Cluster. Ein ECE von 1 bedeutet maximale Entropie. In dem Fall ist der Cluster-Algorithmus nicht in der Lage, die Gruppierung, die aufgrund der Klassenzuordnung erwartet wird, zu erkennen. In dieser Arbeit würde ein ECE von 1 anzeigen, dass die Kultivierungsparameter keinen Einfluss auf die Verteilung der Spektren im Datenraum haben.

2.3.5.2 Topologieerhaltende Karten nach Kohonen

Unter den unüberwachten Lernmethoden stellen die in den 1980er Jahren von Teuvo Kohonen entwickelten "Self Organizing Maps" (SOMs) [139], die auch als Kohonen-Karten oder Kohonen-Netze bezeichnet werden, eines der am häufigsten genutzten Modelle dar. Kohonen-Karten zählen zu den künstlichen neuronalen Netzwerken, die dem Nervensystem

des menschlichen Gehirns nachempfunden sind. Während ein Computer eingehende Informationen normalerweise sequentiell verarbeitet, besteht ein neuronales Netz aus vielen kleinen Einheiten – den Neuronen – die sich untereinander über gerichtete Verbindungen aktivieren. Die Signalverarbeitung verläuft auf diese Weise parallel. Durch das nach biologischem Vorbild entwickelte Signalverarbeitungssystem können Eigenschaften des menschlichen Gehirns wie Selbstorganisation bzw. Lernfähigkeit oder Generalisierungs- bzw. Assoziationsfähigkeit in vereinfachter Form auf den Computer übertragen werden. Neuronale Netze werden je nach Art für verschiedene Problemklassen verwendet. So haben SOMs die Fähigkeit, einen hochdimensionalen Datensatz auf einem niedrig dimensionalen Gitter abzubilden, also eine Karte des hochdimensionalen Raums zu zeichnen. Bei der Projektion bleibt die Topologie des Eingaberaums weitgehend erhalten, was dazu führt, dass sich ähnliche Eingabemuster auch auf der SOM nahe beieinander anordnen. Somit dienen SOMs als Visualisierungsmethode, die die Erkennung von Strukturen, Klassen und Clustern in hochdimensionalen Datenräumen ermöglicht. Abbildung 2.20 illustriert die topologieerhaltende Eigenschaft der SOMs.

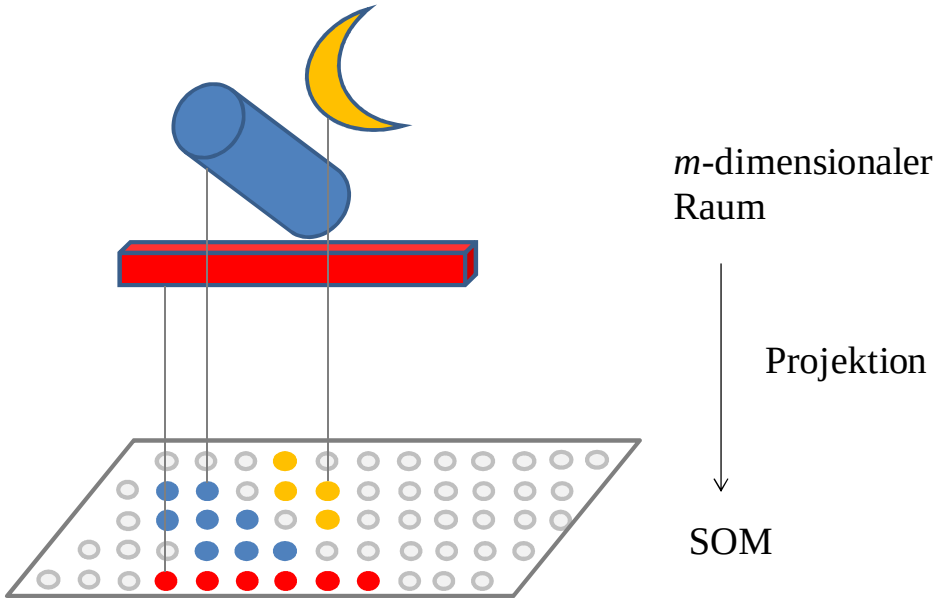


Abbildung 2.20: Topologieerhaltende Eigenschaft der SOMs. Datenpunkte, die im *m*-dimensionalen Raum nahe beieinander liegen, sind auch auf der SOM benachbart.

Prinzipiell unterscheidet man -wie bei anderen neuronalen Netzen- zwei Phasen: die Lernphase und die Ausführungsphase. Zu Beginn der Lernphase wird eine Neuronen-Karte mit einer vom Benutzer definierten Anzahl an Neuronen vorgegeben. Jedes Neuron u_{ij} ist durch einen Gewichtsvektor $\mathbf{v}_{ij}$ charakterisiert, der zufällig initialisiert wird. Die Dimension dieses Vektors entspricht der Dimension der Input-Daten. Der Index (*ij*) dient der eindeutigen Identifizierung der Neuronen sowie ihrer Gewichtsvektoren auf der Karte. Die Lernphase, bei der die Karte mit den Input-Daten trainiert wird, ist ein iterativer Prozess und besteht aus folgenden Schritten:

Algorithmus 2.7: Lernphase einer "Self organizing map"

1. Wähle einen Trainingsvektor **x** aus der Input-Datenmenge und übergebe ihn der Neuronenkarte.
2. Ermittle das Neuron u_{ij}, dessen Gewichtsvektor $\mathbf{v}_{ij}$ am meisten Ähnlichkeit mit dem Trainingsvektor **x** hat. Dieses wird Gewinnerneuron (engl. Best Matching Unit: *BMU*) genannt.
3. Verändere den Gewichtsvektor $\mathbf{v}_{ij}$ des Gewinnerneurons u_{ij} sowie der Nachbarneuronen (auf der Karte) in Richtung **x** (im Input-Datenraum) gemäß der Kohonen-Lernregel (2.92).
4. Wiederhole Schritt 1-3 für eine vorgegebene Anzahl an Iterationen.

Für die Berechnung der Distanzen im Input-Raum und auf der Kohonen-Karte wird in der Regel die euklidische Distanz verwendet. Die Kohonen-Lernregel ist folgendermaßen definiert:

$$\mathbf{v}_{ij}^{\text{neu}} = \mathbf{v}_{ij}^{\text{alt}} + \alpha(t) \cdot \beta(u_{ij}, BMU, t) \cdot (\mathbf{x} - \mathbf{v}_{ij}^{\text{alt}}) \tag{2.92}$$

Dabei werden die Lernrate $\alpha(t)$ und die Nachbarschaftsfunktion $\beta(u_{ij}, BMU, t)$ problemabhängig definiert. Die Lernrate $\alpha(t)$ bestimmt, wie stark der Gewichtsvektor $\mathbf{v}_{ij}$

eines Neurons u_{ij} in jeder Trainingsstufe verändert wird, d.h. wie stark das Neuron „lernt“. $\alpha(t)$ ist eine zeitabhängige Funktion. Dabei steht *t* für die Anzahl der bereits durchgeführten Trainingsschritte. Zu Beginn der Lernphase wird normalerweise ein höherer Wert für die Lernrate angesetzt, der während des Lernprozesses immer kleiner wird. Dadurch entfaltet sich die Neuronen-Karte anfänglich schnell und grob, während bei weiterem Fortschreiten nur noch eine Feinjustierung der Karte stattfindet. Die Nachbarschaftsfunktion $\beta(u_{ij}, BMU, t)$ definiert, welche Neuronen lernen dürfen und in welchem Ausmaß. Neuronen mit großem Abstand zum Gewinnerneuron (gemäß der euklidischen Distanz auf der SOM) lernen weniger stark als Neuronen mit kleinem Abstand zum Gewinnerneuron. Ebenso wie die Lernrate, so ist auch die Nachbarschaftsfunktion zeitabhängig. Zu Beginn wird normalerweise ein großer Nachbarschaftsradius $N(t)$ angesetzt, was bedeutet, dass auch Neuronen die weiter vom Gewinnerneuron entfernt sind, lernen dürfen. Später wird der Nachbarschaftsradius kleiner (siehe Abbildung 2.21). Auf diese Weise kommt es wiederum zu einer Grobanpassung der Karte zu Beginn und zu einem „Fine-Tuning“ zum Ende der Lernphase.

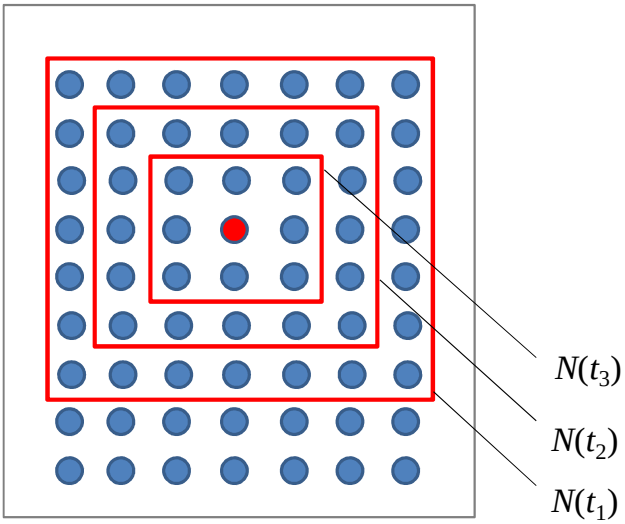


Abbildung 2.21: Zeitabhängige Entwicklung der Nachbarschaftsfunktion $\beta(u_{ij}, BMU, t)$ während des Trainings einer SOM. Das Gewinnerneuron (*BMU*) ist rot gekennzeichnet. Zu Beginn der Trainingsphase ist der Radius der lernenden Neuronen größer ($N(t_1)$) als zu einem späteren Zeitpunkt der Lernphase ($N(t_2)$ und $N(t_3)$).

In der Literatur sind eine Reihe unterschiedlicher Nachbarschaftsfunktionen beschrieben [140]. Ein Beispiel für eine häufig verwendete Nachbarschaftsfunktion ist die Gauß-Funktion (Gl. (2.93)), deren Nachbarschaftsradius $N(t)$ mit Fortschreiten des Lernens exponentiell abnimmt (Gl. (2.94)):

$$\beta(\mathrm{u}_{ij}, BMU, N) = \mathrm{e}^{-\left(\frac{D(u_{ij}, BMU)}{N}\right)^2} \tag{2.93}$$

$$N(t) = N_0 \cdot \mathrm{e}^{-\frac{t}{t_{max}}} \tag{2.94}$$

Dabei beschreibt $D(\mathrm{u}_{ij}, BMU)$ die Distanz (hier euklidische Distanz) eines Neurons u_{ij} vom Gewinnerneuron *BMU*. N_0 ist der Nachbarschaftsradius zum Zeitpunkt Null (zu Beginn der Trainingsphase). t_{max} bezeichnet die Anzahl der maximalen Trainingsschritte.
Während des Lernvorgangs findet eine Art Segmentierung statt, wodurch sich Neuronen mit ähnlichen Gewichtsvektoren innerhalb der Karte nahe beieinander anordnen. Wird das Netz mit einem Trainingsvektor häufiger konfrontiert als mit den restlichen Trainingsvektoren, ist der entsprechende Bereich auf der Neuronen-Karte detaillierter abgebildet. Im Anschluss an die Lernphase werden in der Ausführungsphase die Objekte des *m*-dimensionalen Raums jeweils einem Neuron auf der Karte zugeteilt, was aufgrund der Ähnlichkeit (euklidische Distanz) der Objektvektoren zu den Gewichtsvektoren der Neuronen geschieht.

3 Klassifikation von Reinraumbakterien

Im Rahmen der vorliegenden Dissertation wird ein geeignetes datenanalytisches Verfahren für die Differenzierung von Reinraumbakterien im industriellen „Online-Monitoring“ entwickelt. Dazu werden die im vorangehenden theoretischen Teil beschriebenen Methoden miteinander verglichen, kombiniert und nach ihrer Leistungsfähigkeit für die vorliegende Aufgabenstellung beurteilt. Die experimentellen Daten sowie die erzielten datenanalytischen Ergebnisse werden in den folgenden Abschnitten vorgestellt und diskutiert.

3.1 Experimenteller Aufbau

Nachdem in Kapitel 2.2 bereits die Grundlagen der Raman-Spektroskopie beschrieben wurden, wird in diesem Abschnitt das Prinzip der konfokalen Raman-Spektroskopie, welches hier zum Einsatz kommt, sowie das für die spektroskopischen Messungen verwendete Mikro-Raman-Setup [14] vorgestellt.

3.1.1 Konfokale Mikro-Raman-Spektroskopie

Die konfokale Mikro-Raman-Spektroskopie ist eine Kombination aus Raman-Spektroskopie und konfokaler optischer Laser-Mikroskopie. Sie hat den Vorteil einer hohen Ortsauflösung, so dass sehr kleine Probenmengen wie beispielsweise einzelne Bakterienzellen Raman-spektroskopisch vermessen werden können. Während bei einem konventionellen Mikroskop die gesamte Fläche gleichmäßig belichtet und betrachtet wird, findet bei der konfokalen Anordnung eine Fokussierung des Lasers auf eine kleine Stelle der Probe statt. Nur dieser Fokus wird betrachtet. Dabei laufen folgende Schritte ab. Nach fokussierter Laseranregung passiert das von der Probe ausgehende, Raman-verschobene Licht das Mikroskop-Objektiv (MO) in umgekehrter Richtung und gelangt über den Strahlteiler (ST2) und durch eine Lochblende (Pinhole) in das Spektrometer bzw. den Detektor. Der Anregungsfokus und der

Detektionsfokus liegen dabei aufeinander (konfokal). Alle Strahlen, die nicht aus dem Fokus kommen, werden durch die Lochblende eliminiert (siehe Abbildung 3.1).

Durch die konfokale Mikro-Raman-Spektroskopie erhöht sich im Vergleich zur konventionellen Raman-Spektroskopie sowohl die laterale Auflösung (Auflösung bezüglich zweier nebeneinanderliegender Punkte) als auch die axiale Auflösung (Tiefenschärfe, Auflösung entlang der optischen Achse).

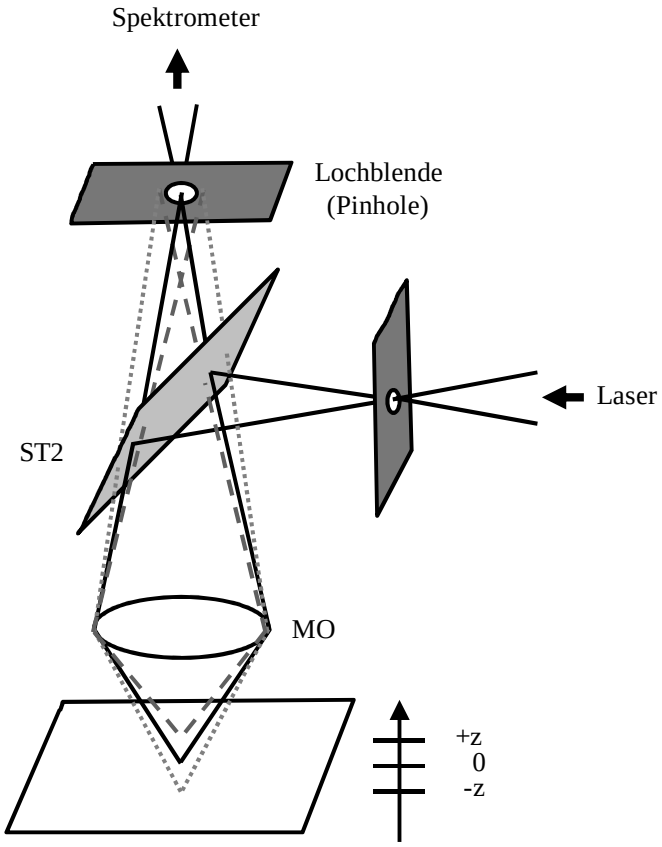


Abbildung 3.1: Konfokales Mikro-Raman-Setup (aus [14] ins Deutsche übersetzt, Copyright Wiley-VCH Verlag GmbH & Co. KGaA, mit freundlicher Genehmigung des Wiley-Verlags). Nur Raman-Strahlung, die an der fokussierten Stelle der Probe aufgenommen wird (durchgezogene Line), gelangt über den Strahlenteiler (ST2) und das Pinhole ins Spektrometer. Strahlen die außerhalb des Fokus liegen (durchbrochene Linien), werden von der Lochblende abgefangen und gelangen nicht in das Spektrometer.

3.1.2 Mikro-Raman-Setup

Die in der vorliegenden Arbeit analysierten Spektren wurden mit Hilfe eines Mikro-Raman-Spektrometers (HR LabRam Invers, Jobin-Yvon-Horiba) im spektralen Bereich von 537 cm^{-1} bis 3654 cm^{-1} mit einer Integrationszeit von 60 Sekunden aufgenommen (siehe Abbildung 3.2). Dazu wurden die Bakterienzellen auf einem Objektträger ausgestrichen. Die Anregung erfolgte durch einen frequenzverdoppelten Nd:YAG Laser (Coherent Compass) bei 532 nm mit einer Laserleistung von ungefähr 2.4 mW.

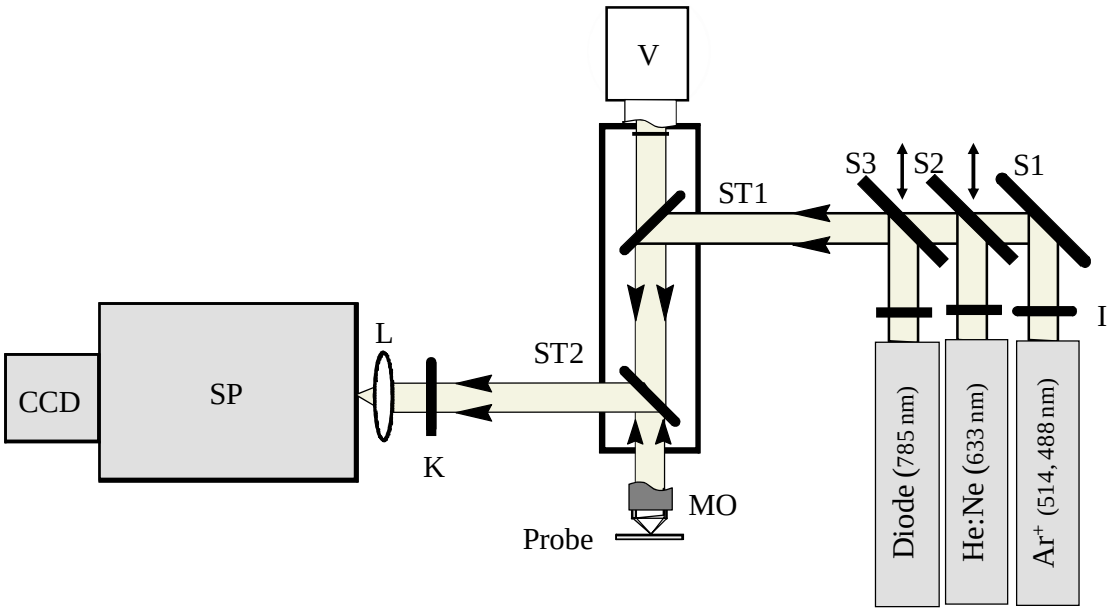


Abbildung 3.2: Schematische Darstellung des Mikro-Raman-Setup (aus [14] ins Deutsche übersetzt, Copyright Wiley-VCH Verlag GmbH & Co. KGaA, mit freundlicher Genehmigung des Wiley-Verlags): I: Interferenzfilter, S1-3: Spiegel für verschiedene Anregungswellenlängen, ST1, ST2: Strahlenteiler, V: Video-Kontrolle, MO: Mikroskop-Objektiv, K: Kerbfilter, L: Linse, SP: Spektrometer, CCD: Charge Coupled Devise Detektor.

Bei dem verwendeten Mikro-Raman-Setup, das in Abbildung 3.2 dargestellt ist, wird der Laserstrahl über ein System von Spiegeln (S1, S2, S3) in das Mikroskop geleitet. Ein Interferenzfilter (I) entfernt unerwünschte Nebenfrequenzen des Laserlichtes. Ein Mikroskop-Objektiv (MO) dient dazu, den Laser auf einen Durchmesser von ungefähr

0.7 μm zu fokussieren und das nach der Einstrahlung entstehende Streulicht aufzunehmen. Die elastisch gestreuten Lichtanteile werden durch einen Kerbfilters (K) entfernt. Das restliche Licht wird über eine Linse (L) auf die Eintrittsöffnung (bzw. Pinhole) des Spektrometers (SP) fokussiert und über eine CCD Kamera (engl. Charge-Coupled Device Camera), die auf 220 K temperiert ist, detektiert. Eine optische Überwachung der Probe ist durch eine Kamera (V) möglich. Die Kalibrierung des Spektrometers wird als täglicher Routine-Check mit Titandioxid als Referenz durchgeführt.

3.2 Untersuchte Bakterien

Die Studie basiert auf einem Datensatz, der 29 verschiedene Bakterienstämme und 3642 Spektren enthält [14]. Da ein Verfahren für das „Online Monitoring“ in industriellen Reinräumen entwickelt werden soll, wurden die Bakterienstämme nach ihrem Vorkommen in industriellen Reinräumen ausgewählt (siehe Tabelle 3.1).

Die Stämme wurden von der Deutschen Sammlung von Mikroorganismen und Zellkulturen (DSMZ) in Braunschweig sowie vom Institut für Infektionsbiologie der Universität Würzburg bezogen. Um Einflüsse von verschiedenen Wachstumsbedingungen auf die Differenzierbarkeit der Bakterien abschätzen zu können, wurden die Bakterien unter verschiedensten Bedingungen bezüglich Medium, Temperatur und Wachstumsdauer kultiviert. Die verwendeten Kultivierungsmedien sind Nähragar (NA), Standard-1-Nähragar (S-1-NA), Corynebacterium Agar (CA) und Tryptikase-Soja-Hefe-Extrakt-Agar (CASO). Die genaue Datensatzzusammensetzung sowie die verwendeten Kultivierungsbedingungen sind in Tabelle 3.1 aufgeführt. Zusätzlich zu diesem Datensatz wird ein "verblindetes" Testset analysiert (d.h. die Bakterienzugehörigkeit ist während der Datenanalyse und Vorhersage nicht bekannt). Das „verblindete“ Testset besteht aus 16 Bakterienstämmen, die im Trainingsdatensatz enthalten sind, und 6 Bakterienstämmen, die nicht im Trainingsdatensatz vorkommen. Die genaue Zusammensetzung des „verblindeten“ Testdatensatzes ist in Tabelle 3.2 beschrieben. Für die Durchführung der Raman-Messungen an Einzelbakterien, wurden die Bakterien nach der Kultivierung von den Agarplatten entnommen und auf einem Objektträger ausgestrichen.

Tabelle 3.1: Zusammensetzung des untersuchten Bakteriendatensatzes

Name	Kultivierungs-medium	Temperatur	Anzahl der Spektren	Name	Kultivierungs-medium	Temperatur	Anzahl der Spektren
B. pumilus DSM27	NA	30°C	57	*S. epidermidis* ATCC 35984	CASO	37°C	280
B. pumilus DSM 361	NA	30°C	69		CA	37°C	261
B. sphaericus DSM 28	NA	30°C	53		CA	30°C	264
B. sphaericus DSM 396	NA	30°C	42	*S. warneri* DSM 20036	CASO	37°C	22
B. subtilis DSM 10	NA	30°C	326		CA	37°C	21
B. subtilis DSM 347	NA	30°C	42		CA	30°C	22
M. luteus DSM 20030	NA	30°C	48	*S. warneri* DSM 20316	CASO	37°C	22
M. luteus DSM 348	NA	30°C	619		CA	37°C	24
M. lylae DSM 20315	CA	37°C	45		CA	30°C	21
M. lylae DSM 20318	CA	37°C	20	*E. coli* DSM 1058	S-1-NA	37°C	15
S. cohnii DSM 20260	CA	37°C	21		S-1-NA	30°C	23
	CASO	37°C	19		NA	37°C	13
	CASO	30°C	24		NA	30°C	17
S. cohnii DSM 6669	CA	37°C	21	*E. coli* DSM 2769	S-1-NA	37°C	29
	CASO	37°C	20		S-1-NA	30°C	27
	CASO	30°C	21		NA	37°C	29
S. cohnii DSM 6718	CA	37°C	21		NA	30°C	23
	CASO	37°C	19	*E. coli* DSM 423	S-1-NA	37°C	25
	CASO	30°C	21		S-1-NA	30°C	16
S. cohnii DSM 6719	CA	37°C	21		NA	37°C	50
	CASO	37°C	18		NA	30°C	21
	CASO	30°C	22	*E. coli* DSM 429	S-1-NA	37°C	25
S. epidermidis DSM 1798	CASO	37°C	22		S-1-NA	30°C	20
	CASO	30°C	38		NA	37°C	27
	CA	37°C	23		NA	30°C	18
	CA	30°C	29	*E. coli* DSM 498	S-1-NA	37°C	26
S.epidermidis 195	CASO	37°C	17		S-1-NA	30°C	16
	CASO	30°C	18		NA	37°C	21
	CA	37°C	22		NA	30°C	23
	CA	30°C	17	*E. coli* DSM 499	S-1-NA	37°C	23
S. epidermidis DSM 20042	CASO	37°C	23		S-1-NA	30°C	18
	CASO	30°C	27		NA	37°C	20
	CA	37°C	20		NA	30°C	22
	CA	30°C	36	*E. coli* DSM 613	S-1-NA	37°C	23
S. epidermidis DSM 3269	CASO	37°C	24		S-1-NA	30°C	26
	CASO	30°C	22		NA	37°C	21
	CA	37°C	23		NA	30°C	24
	CA	30°C	24				
S. epidermidis DSM 3270	CASO	37°C	23				
	CASO	30°C	36				
	CA	37°C	24				
	CA	30°C	27				

Tabelle 3.2: Zusammensetzung des „verblindeten“ Testdatensatzes

Name		**Anzahl der Spektren**
B. sphaericus	DSM 28	8
B. sphaericus	DSM 396	7
B. subtilis	DSM 347	8
M. luteus	DSM 20030	6
M. lylae	DSM 20315	5
M. lylae	DSM 20318	5
S. cohnii	DSM 20260	7
S. cohnii	DSM 6669	8
S. cohnii	DSM 6718	5
S. cohnii	DSM 6719	5
S. epidermidis	DSM 44195	20
S. epidermidis	ATCC 35984	7
S. warneri	DSM 20036	5
E. coli	DSM 1058	20
E. coli	DSM 423	7
E. coli	DSM 498	7
Anzahl der “bekannten” Bakterienstämme		**130**
M. luteus	BCD 3906	45
E. coli	DSM 5208	26
E. coli	DSM 426	24
S. hominis	BCD 2684	21
S. thermophilus	DSM 20617	28
L. acidophilus	DSM 9126	25
Anzahl der “unbekannten” Bakterienstämme		**169**

3.3 Software

Alle datenanalytischen Methoden wurden mit MATLAB R2006b und R2007a (The MathWorks, Natick, MA, USA) durchgeführt. Für PLS-DA und *k*NN wurde die PLS-Toolbox (Eigenvector Research) verwendet. Zur Berechnung von LDA und QDA diente die "Statistics Toolbox" von MATLAB. MDA wurde mit einer selbstgeschriebenen MATLAB-Routine berechnet (siehe Anhang B). Für SVMs wurde das von Chang und Lin entwickelte MATLAB-Interface der LIBSVM (Library for „Support Vector Machines“) [89] verwendet. Die SOM Toolbox, die von Alhoniemi et al. für MATLAB entwickelt wurde, diente zum Training der SOMs. Für die Signifikanztests wurden die Funktionen VARTESTN, ANOVA1, ANOVA2, KRUSKALWALLIS und MULTCOMPARE von MATLAB´s

"Statistics Toolbox" herangezogen. Zur Berechnung der Multi-Klassen *a posteriori* Wahrscheinlichkeiten bei der paarweisen Klassifikation (siehe Kapitel 2.3.3.5.1) wurden MATLAB-Funktionen von Wu et al. verwendet, die im Zusammenhang mit der Publikation [86] unter [141] abgerufen werden können.

3.4 Ergebnisse und Diskussion

3.4.1 Datenvorbehandlung

Der 1. Schritt der Datenauswertung besteht darin, eine für die Raman-Spektren geeignete Vorbehandlungsmethode zu finden. In Abbildung 2.5A sind Beispiele für Rohspektren aller 29 Bakterienstämme dargestellt. Wie in Kapitel 2.3.1 beschrieben, wurden die Spektren vor der tatsächlichen Datenauswertung durch Interpolation auf ein einheitliches Wellenzahlspektrum gebracht (INTERPOL) und die enthaltenen „Spikes“ eliminiert. Diese beiden Arten der Vorbehandlung werden hier als grundlegend angesehen, da sowohl Verschiebungen in den Messpunkten, an denen die Raman-Intensitäten aufgenommen wurden, als auch „Spikes“ die Klassifikation stark beeinträchtigen können. Die interpolierten und „Spike“-eliminierten Spektren, die als SPIKEELIM gekennzeichnet sind, wurden mit den im theoretischen Teil vorgestellten Methoden der Normierung (VEKNORM) und Basislinienkorrektur (1. ABL, POLY4, WHIT) behandelt (siehe Abbildung 2.6).

Im Folgenden wird zunächst der Einfluss der Vorbehandlungstechniken auf den Klassifikationserfolg beschrieben (siehe Kapitel 3.4.1.1). Die so erhaltenen Ergebnisse führten dazu, dass zudem die Methoden der Basisliniensubtraktion genauer analysiert wurden, was in Kapitel 3.4.1.2 beschrieben ist. Einige schwingungsspektroskopische Studien schlagen eine Kombination von Basislinienkorrektur und Normierung vor, um die Robustheit der Auswertung zu erhöhen [142]. Dies wird hier ebenso diskutiert und ist Gegenstand von Kapitel 3.4.1.3.

3.4.1.1 Vergleich der Vorbehandlungsmethoden

3.4.1.1.1 Ergebnisse

In Abbildung 3.3 sind die mittleren Wiedererkennungsraten verschiedener Klassifikations- und Vorbehandlungsmethoden unter Verwendung der 50-fachen Kreuzvalidierung aufgetragen.

Die Signifikanz der numerischen Unterschiede zwischen den Vorhersageergebnissen wurde in dieser Studie mit Hilfe von ein- und zweifaktorieller CVANOVA bewertet. Die detaillierten theoretischen Grundlagen einer CVANOVA sind in Kapitel 2.3.4.2 beschrieben. Für jede Methodenkombination (Klassifikations- und Vorbehandlungsmethode) sind 50 Einzelvorhersagewerte (Wiedererkennungsraten) vorhanden, da eine 50-fache Kreuzvalidierung durchgeführt wurde. Bei einer CVANOVA wird die Varianz der Einzelvorhersagewerte pro Methodenkombination mit den Varianzen, die zwischen den Methodenkombinationen bestehen, ins Verhältnis gesetzt. Dadurch können Aussagen über die Signifikanz der Unterschiede zwischen den Klassifikationsergebnissen gemacht werden.

Um zu überprüfen, ob die Voraussetzungen zur Durchführung einer CVANOVA erfüllt sind, wurde -wie in Kapitel 2.3.4.2.1 beschrieben- auf Normalverteilung und Varianzhomogenität der Einzelvorhersagewerte getestet. Für diesen Zweck stellen die visuelle Inspektion von Normal-Quantil-Quantil-Plots (Normal-Q-Q-Plots) [125] und der Bartlett´s Test [129] geeignete Methoden dar. Die Theorie zur Bildung von Normal-Q-Q-Plots sowie die in dieser Arbeit erhaltenen Normal-Q-Q-Plots sind in Anhang A der Arbeit zu finden. Bei der Betrachtung der Normal-Q-Q-Plots (siehe Abbildungen A.1-A.6) stellte sich heraus, dass das Kriterium der Normalverteilung für die meisten Methodenkombinationen annähernd erfüllt ist. Stärkere Abweichungen beobachtete man nur vereinzelt bei den nicht-parametrischen Klassifikationsmethoden *k*NN und SVM (siehe Abbildungen A.5 und A.6). Bei Prüfung auf Varianzhomogenität mit dem Bartlett´s Test wurde unter Berücksichtigung aller Methodenkominationen (Klassifikations- und Vorbhandlungsmethoden) zunächst festgestellt, dass die Nullhypothese der Varianzhomogenität mit einer Irrtumswahrscheinlichkeit von 5% abgelehnt wird. Bei genauerer Betrachtung stellte sich heraus, dass innerhalb der parametrischen (PLS-DA, LDA, QDA, MDA) sowie innerhalb der nicht-parametrischen Methoden (*k*NN, SVM) annähernd Varianzhomogenität besteht. Zwischen parametrischen

und nicht-parametrischen Methoden kommen aber Varianzinhomogenitäten vor. Dies lässt sich dadurch erklären, dass die nicht-parametrischen Methoden bei bestimmten Vorbehandlungsmethoden neben einer besseren Klassifikationsrate auch eine niedrigere Varianz der Wiedererkennungsraten zeigen als die parametrischen Methoden. Insgesamt wurden die Ergebnisse der Tests in dieser Arbeit aber als ausreichend für eine Durchführung der CVANOVA bewertet, da sich ANOVA sowie der dazu eng verwandte t-Test in der Regel stabil gegenüber nicht zu groben Abweichungen von Normalverteilung und Varianzhomogenität verhalten. Besonders bei einer ausreichenden Anzahl an Stichproben und bei gleichen Stichprobenumfängen, wie es hier der Fall ist, beobachtet man eine hohe Robustheit der ANOVA. Um dennoch Fehler kontrollieren zu können, die durch Abweichungen von den Kriterien der Normalverteilung sowie Varianzhomogenität entstehen, wurden neben der einfaktoriellen CVANOVA zusätzlich Kruskal-Wallis-Tests durchgeführt (siehe Tabelle 3.3). Dabei wurde deutlich, dass der parametrische Ansatz (CVANOVA) sich von dem nicht-parametrischen (Kruskal-Wallis) kaum unterscheidet. So fielen von den 90 paarweisen Vergleichen nach Tukey, die in Tabelle 3.3 beschrieben sind, nur 2 Vergleiche (mit Sternchen gekennzeichnet) für parametrische und nicht-parametrische Tests unterschiedlich aus.

Bei der im Folgenden beschriebenen Analyse mit ein- und zweifakterieller CVANOVA in Kombination mit Tukey´s Test wurden die Unterschiede zwischen den verschiedenen Methodenkombinationen als signifikant angesehen, wenn der α-Wert (Irrtumswahrscheinlichkeit) kleiner als 0.05 ist. Für die CVANOVA bedeutet ein signifikantes Ergebnis, dass die Wahrscheinlichkeit, dass mindestens eine Technik eine Wiedererkennungsrate liefert, die unterschiedlich zu der einer anderen Technik ist, 95% beträgt. Im Anschluss an ein signifikantes Ergebnis der CVANOVA können einzelne Unterschiede zwischen den Methodenkombinationen mit Tukey´s Test festgestellt werden.

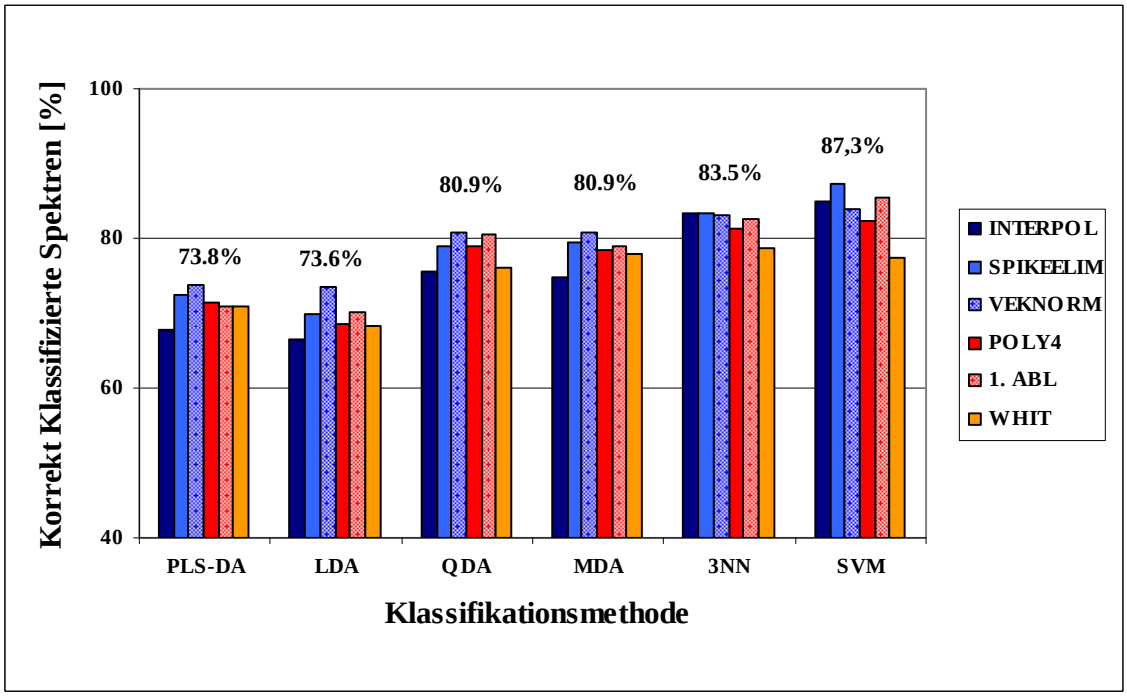


Abbildung 3.3: Wiedererkennungsraten für verschiedene Kombinationen aus Klassifikations- und Vorbehandlungsmethoden. Für die jeweils beste Kombination aus Vorbehandlungs- und Klassifikationsmethode ist der Zahlenwert angegeben.

Die Bewertung der Unterschiede zwischen den in Abbildung 3.3 gezeigten Klassifikationsraten erfolgte zunächst mit Hilfe einer zweifaktoriellen CVANOVA. Dabei wurde simultan der Einfluss von Vorbehandlungs- und Klassifikationsmethoden auf das Klassifikationsergebnis betrachtet. Für beide Gruppen (Vorbehandlungs- und Klassifikationsmethoden) wurden signifikante Unterschiede gefunden. Daraufhin wurde als "Post-Hoc"-Analyse Tukey´s Test (siehe Kapitel 2.3.4.2.4) herangezogen, um auf spezifische paarweise Unterschiede zwischen den einzelnen Vorbehandlungsmethoden zu prüfen. Die Ergebnisse von Tukey´s Test sind in Abbildung 3.4 gezeigt und werden im Diskussionsteil (Kapitel 3.4.1.1.2) besprochen.

Neben signifikanten Unterschieden wurden durch die CVANOVA auch signifikante Interaktionen zwischen den Gruppen gefunden. Das bedeutet, dass die Wiedererkennungsrate bezüglich einer Vorbehandlungsmethode von der verwendeten Klassifikationsmethode

abhängt. Deshalb zeigen die Ergebnisse, die bei Tukey´s Test erhalten wurden (siehe Abbildung 3.4) nur einen generellen Trend der Daten. Diese Effekte werden als Haupteffekte (engl. Main Effects) bezeichnet. Erweist sich eine Vorbehandlungsmethode in den Haupteffekten als signifikant besser als die anderen Methoden, gilt dies also nicht notwendigerweise für alle Klassifikationsmethoden. Um herauszufinden, inwieweit sich die Methodenkombinationen voneinander unterscheiden, wurde deshalb für jede einzelne Klassifikationsmethode zusätzlich eine einfaktorielle CVANOVA mit anschließendem Tukey´s Test durchgeführt. Nach Tukey´s Test konnte nun innerhalb der Klassifikationsmethoden festgestellt werden, welche Unterschiede zwischen den Vorbehandlungsmethoden bestehen. Diese Effekte werden als einfache Effekte (engl. Simple Effects) bezeichnet und sind in Tabelle 3.3 gezeigt.

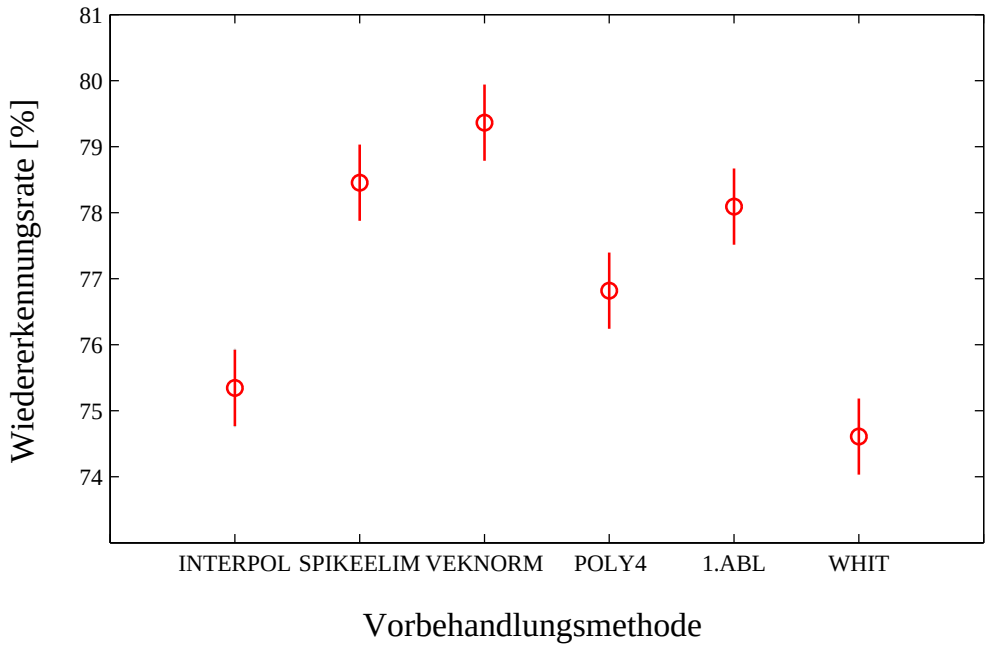


Abbildung 3.4: Im Anschluss an eine signifikante zweifaktorielle CVANOVA, wurde Tukey´s Test verwendet, um signifikante Unterscheide (Haupteffekte) zwischen den Wiedererkennungsraten der verschiedenen Vorbehandlungsmethoden zu finden. Die Kreise kennzeichnen die durchschnittliche Wiedererkennungsrate der verschiedenen Vorbehandlungsmethoden (alle Klassifikationsmethoden werden dabei gleichzeitig berücksichtigt). Die Balken beschreiben die Konfidenzintervalle. Überlappende Konfidenzintervalle zeigen an, dass sich die Klassifikationsraten nicht signifikant unterscheiden ($\alpha < 0.05$).

Tabelle 3.3: Aufgrund von statistisch signifikanten Interaktionen zwischen Klassifikations- und Vorbehandlungsmethoden sind die Haupteffekte nicht für alle Methodenkombinationen repräsentativ. Die Signifikanz der Unterschiede zwischen den Vorbehandlungsmethoden wurden daraufhin auf der Basis der einzelnen Klassifikationsmethoden durch eine einfaktorielle CVANOVA in Kombination mit Tukey´s Test analysiert (einfache Effekte). Signifikante Unterschiede zwischen zwei Vorbehandlungsmethoden bezüglich einer Klassifikationsmethode sind durch einen Eintrag in der Tabelle gekennzeichnet. Zusätzlich wurde der nicht-parametrische Kruskal-Wallis-Test in Verbindung mit Tukey´s Test angewendet. Ein Sternchen (*) neben einem Eintrag in der Tabelle kennzeichnet Unterschiede zwischen der parametrischen (CVANOVA+Tukey) und der nicht-parametrischen (Kruskal-Wallis+Tukey) Methode. Dies bedeutet jeweils, dass mit der parametrischen Methode ein signifikanter Unterschied festgestellt wurde und mit der nicht-parametrischen Methode nicht.

	PLS-DA	**LDA**	**QDA**
INTERPOL	SPIKEELIM, VEKNORM, POLY4, 1.ABL*	SPIKEELIM, VEKNORM, 1.ABL	SPIKEELIM, VEKNORM, POLY4, 1.ABL
SPIKEELIM	INTERPOL	INTERPOL, VEKNORM	INTERPOL
VEKNORM	INTERPOL, WHIT	INTERPOL, SPIKEELIM, POLY4, 1.ABL,WHIT	INTERPOL, WHIT
POLY 4	INTERPOL	VEKNORM	INTERPOL
1. ABL	INTERPOL*	INTERPOL, VEKNORM	INTERPOL,WHIT
WHIT	VEKNORM	VEKNORM	VEKNORM, 1.ABL
	MDA	***k*NN**	**SVM**
INTERPOL	SPIKEELIM, VEKNORM, POLY4, 1.ABL	WHIT	POLY4*, WHIT
SPIKEELIM	INTERPOL	WHIT	VEKNORM, POLY4, WHIT
VEKNORM	WHIT, INTERPOL	WHIT	SPIKEELIM, WHIT
POLY 4	INTERPOL		INTERPOL*, SPIKEELIM, 1.ABL, WHIT
1. ABL	INTERPOL	WHIT	WHIT, POLY4
WHIT	VEKNORM	INTERPOL, SPIKEELIM, VEKNORM, 1.ABL	INTERPOL, SPIKEELIM, VEKNORM, POLY4, 1.ABL

3.4.1.1.2 Diskussion

Betrachtet man zunächst die Haupteffekte der Vorbehandlungsmethoden, so fällt auf, dass die „Spike“-Eliminierung (SPIKEELIM) im Vergleich zu den interpolierten Spektren (INTERPOL) signifikant die Wiedererkennungsrate erhöht (siehe Haupteffekte in Abbildung 3.4). Die Raman-Spektren sollten also grundsätzlich vor der Analyse einer „Spike“-Eliminierung unterzogen werden. Obwohl viele Spektren durch einen starken Fluoreszenzhintergrund gestört sind, konnte die Wiedererkennungsrate durch die Methoden der Basislinienkorrektur nicht verbessert werden. So liefern sowohl der robuste Polynomfit (POLY4) als auch die Glättung nach Whittaker (WHIT) signifikant schlechtere Ergebnisse als die interpolierten und „Spike“-eliminierten Rohspektren (SPIKEELIM) (siehe Haupteffekte in Abbildung 3.4). Vor allem bei der Klassifikation mittels SVMs (siehe Abbildung 3.3 und einfache Effekte in Tabelle 3.3) führt die Basislinienkorrektur zu deutlich schlechteren Vorhersagewerten, während die parametrischen Klassifikationsmethoden nur leicht reduzierte Klassifikationsraten zeigen. Vektornormierung (VEKNORM), bei der die Originalform der Spektren erhalten bleibt, ist den Methoden der Basislinienkorrektur signifikant überlegen (siehe Haupteffekte in Abbildung 3.4). Gegenüber den „Spike“-eliminierten Rohspektren ist das bessere Abschneiden der Vektornormierung in den Haupteffekten allerdings nicht signifikant. So konnte nur für LDA eine signifikante Steigerung der Wiedererkennungsrate im Vergleich zu den „Spike“-eliminierten Rohspektren festgestellt werden (siehe einfache Effekte in Tabelle 3.3). Für die nicht-parametrischen Klassifikationsansätze (*k*NN und SVMs) beobachtet man sogar eine Verschlechterung der Vorhersagegenauigkeit durch die Vektornormierung (siehe Abbildung 3.3 und einfache Effekte in Tabelle 3.3). Insgesamt fällt bei der Betrachtung der einfachen Effekte auf, dass sich parametrische und nicht-parametrische Methoden bezüglich der Vorbehandlungsmethoden unterschiedlich verhalten. So führt die Vektornormierung bei parametrischen Methoden -wie bereits erwähnt- zu einer besseren Klassifikationsrate. Durch Basislinienkorrektur nimmt die Vorhersagegenauigkeit nur leicht (nicht signifikant) ab. Bei den nicht-parametrischen Methoden hingegen führt jede Art der Spektrenvorbehandlung zu einem Verlust an Vorhersagekraft. Alle Methoden der Normierung und Basislinienkorrektur außer der 1. Ableitung zeigen bei den SVMs signifikant schlechtere Klassifikationsraten als die „Spike“-eliminierten Spektren (SPIKEELIM).

Die Ergebnisse lassen vermuten, dass in den Raman-Spektren neben additiven Effekten auch multiplikative Effekte eine Rolle spielen. Additive Effekte sind in den Spektren in Form von horizontal versetzten und steigenden bzw. fallenden Basislinien deutlich erkennbar. Die Korrektur dieser Effekte durch die Methoden der Basislinienkorrektur führt in dieser Studie allerdings zu keiner verbesserten Vorhersagekraft. Multiplikative Effekte sind in den Spektren mit dem Auge nicht erkennbar. Da man allerdings mit vektornormierten Spektren bessere Ergebnisse erhält, als mit unbehandelten oder basislinienkorrigierten Spektren (gilt nur für parametrische Methoden!), ist es hier naheliegend, dass auch multiplikative Effekte vorhanden sind. Diese entstehen möglicherweise durch Größenunterschiede zwischen den Bakterien eines Stammes. Die Tatsache, dass sich eine Basisliniensubtraktion bei allen Methoden negativ auf die Klassifikationsergebnisse auswirkt (nur bei den nicht-parametrischen Methoden signifikant), führt zu der Annahme, dass durch die Basislinienkorrektur fälschlicherweise spektrale Information entfernt wird. Das bedeutet, dass möglicherweise Peaks bei der Korrektur in Mitleidenschaft gezogen werden. In diesem Fall ist die gewählte Methode für die Basislinienschätzung nicht geeignet oder die Parameter der Vorbehandlungsmethode wurden schlecht eingestellt. Die Ursache kann aber auch darin liegen, dass Klasseninformation in der Basislinie vorhanden ist, die für die Klassifikation von Nutzen ist. Diese Überlegungen werden im folgenden Kapitel (Kapitel 3.4.1.2) nochmals aufgegriffen und diskutiert.

Aufgrund der bisherigen Ergebnisse kann man festhalten, dass für parametrische Klassifikationsmethoden (PLS-DA, LDA, QDA, MDA) die Vektornormierung (VEKNORM) die besten Ergebnisse zeigt, während für nicht-parametrische Klassifikationsmethoden (SVM, 3NN) mit den „Spike“-eliminierten Rohspektren die besten Wiedererkennungsraten erhalten werden. Diese Arten der Vorbehandlung wurden für alle weiteren Klassifikationen verwendet, soweit nicht andere Parameter angegeben sind.

3.4.1.2 Studie zur Basisliniensubtraktion

Der Fluoreszenzhintergrund wird in der Raman-Spektroskopie normalerweise als Störfaktor betrachtet, weshalb vor der Analyse eine Basislinienkorrektur durchgeführt wird. In dieser Arbeit nehmen jedoch die Wiedererkennungsraten aller Klassifizierer durch eine

Basislinienkorrektur ab (siehe Abbildung 3.3 und Abbildung 3.4), was besonders bei nicht-parametrischen Methoden (*k*NN, SVMs) beobachtet wird (siehe Abbildung 3.3 und einfache Effekte in Tabelle 3.3). Bei der Verwendung der 1. Ableitung ist der beschriebene Effekt wesentlich schwächer ausgeprägt. So führt die 1. Ableitung nur zu einem leichten Verlust an Vorhersagegenauigkeit. Starke Effekte registriert man dagegen bei den Methoden, die direkt auf der Schätzung und Subtraktion der Basislinie beruhen (POLY4, WHIT), was zu der Annahme führt, dass die Basislinien für die Klassifikation relevante Informationen enthalten. Um dieser Vermutung nachzugehen, wurden die folgenden zwei Tests durchgeführt:

- Neben einer Klassifikation mit Hilfe der basislinienkorrigierten Spektren wurden die Bakterienstämme allein auf Basis der geschätzten Basislinien klassifiziert.
- Es wurden nur ausgewählte Wellenzahlen für die Klassifikation verwendet. Wellenzahlen, die spektrale Information (Peaks) enthalten, gingen in die Berechnung ein, wohingegen Wellenzahlen die nur Basislinie enthalten, entfernt wurden. Die für die Klassifikation verwendeten Wellenzahlen sind 850-1750 cm^{-1} und 2650-3150 cm^{-1}.

Für den ersten Test, wurden die Spektren mit einem robusten Polynomfit verschiedener Ordnung (1., 2., 4., 6., 8. Ordnung) sowie mit dem Whittaker-Algorithmus vorbehandelt. Anschließend wurden sowohl die korrigierten Spektren als auch die geschätzten Basislinien für die Klassifikation herangezogen. Um einschätzen zu können, inwieweit sich die Spektren bei der Basisliniensubtraktion verändern, sind die berechneten Basislinien und die basislinienkorrigierten Spektren der verschiedenen Polynome und des Whittaker-Algorithmus in Abbildung 3.6A-F dargestellt. Die „Spike“-eliminierten Originalspektren vor der Basislinienkorrektur zeigt Abbildung 3.5. Die jeweiligen Ergebnisse der Klassifikation findet man in Abbildung 3.7.

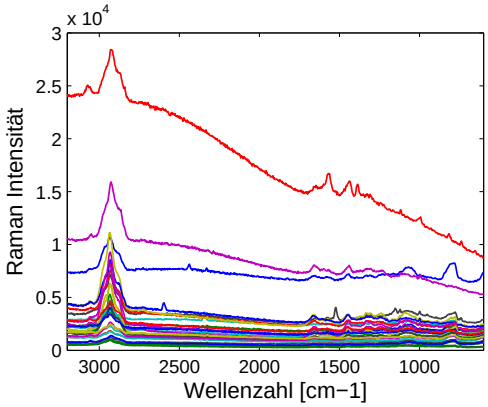
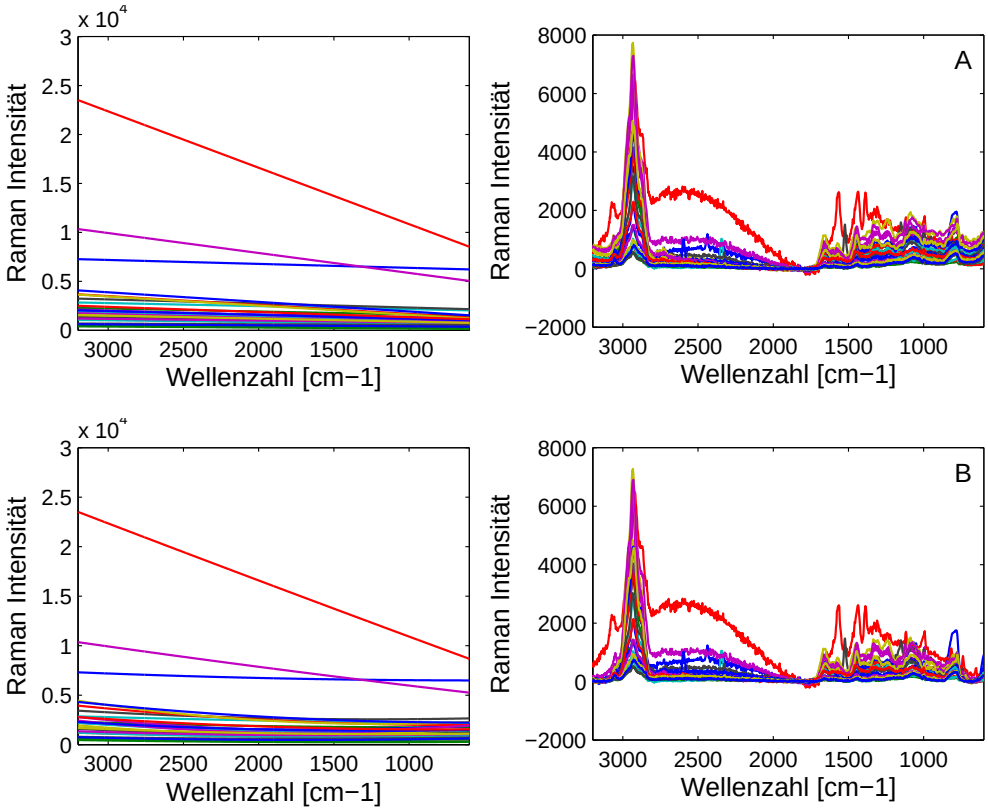
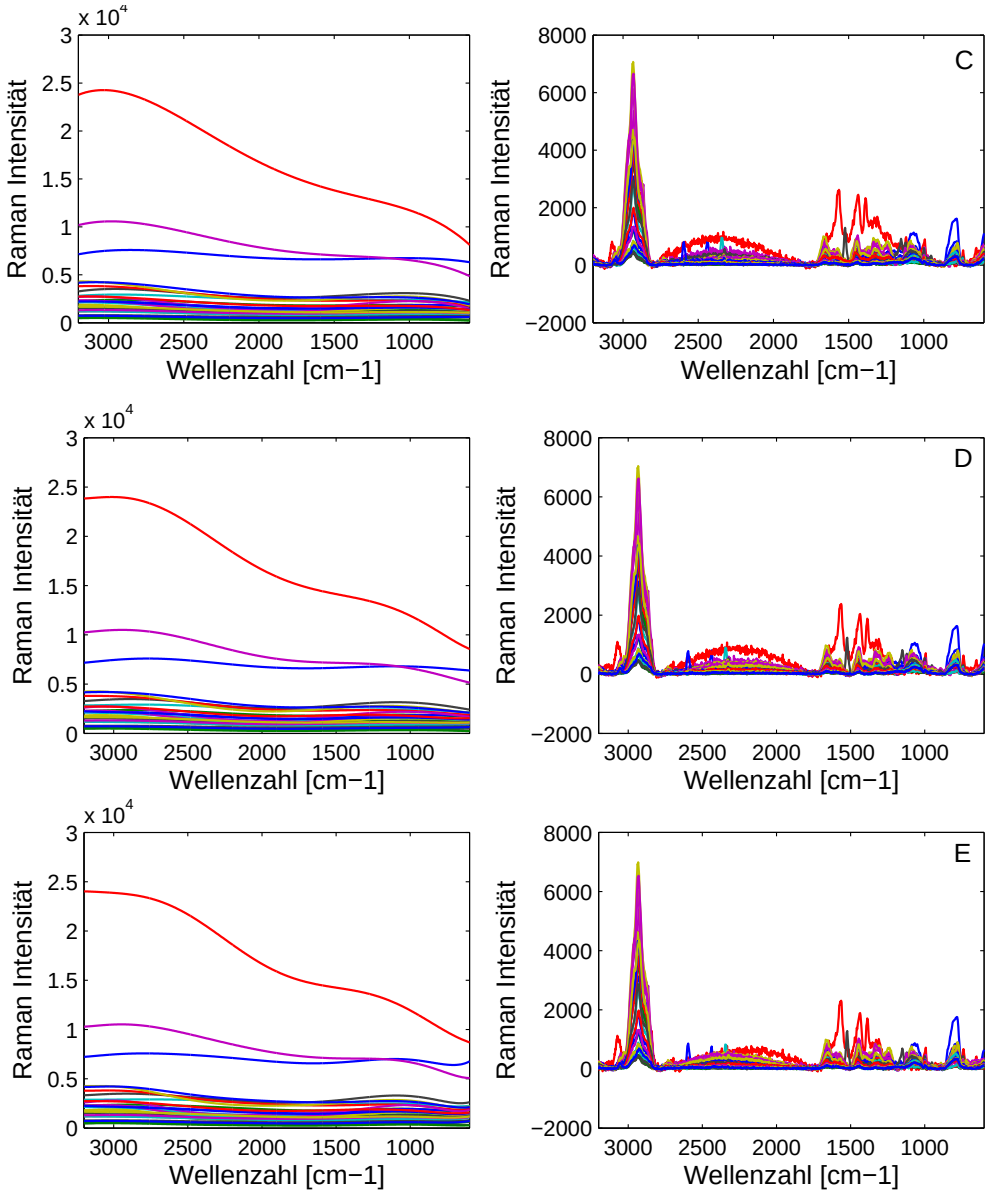


Abbildung 3.5: Interpolierte und „Spike“-eliminierte Raman-Spektren vor der Basislinienkorrektur.




x 10^4
Raman Intensität
Wellenzahl [cm−1]
C
D
E

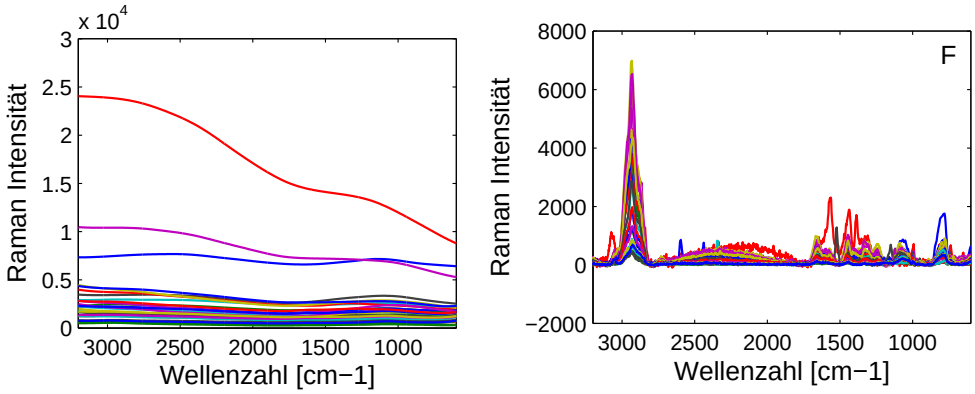


Abbildung 3.6A-E: geschätzte Basislinien sowie basislinienkorrigierte Spektren bei Verwendung des robusten Polynomfit verschiedener Ordnung (**A**: Polynom 1. Ordnung **B**: Polynom 2. Ordnung **C**: Polynom 4. Ordnung **D**: Polynom 6. Ordnung **E**: Polynom 8. Ordnung) **F**: geschätzte Basislinien sowie basislinienkorrigierte Spektren bei Verwendung des Whittaker-Algorithmus.

3.4.1.2.1 Ergebnisse

Abbildung 3.7 zeigt die Klassifikationsraten sowohl der basislinienkorrigierten Spektren (A) als auch der Basislinien selbst (B). In Tabelle 3.4 sind die Ergebnisse der Klassifikation des gesamten Spektrums (SPIKEELIM) sowie der Klassifikation mit ausgewählten Wellenzahlen notiert.

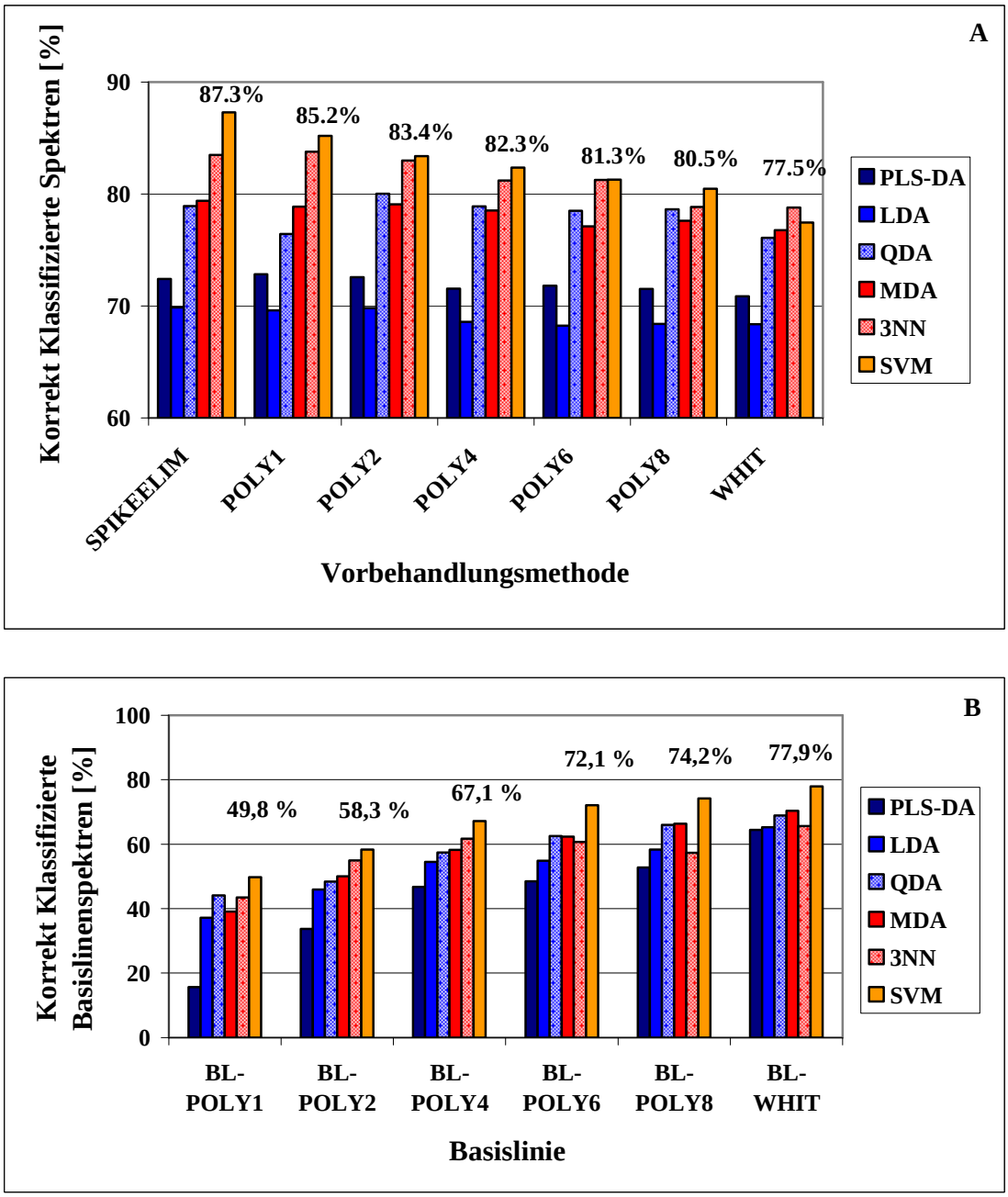


Abbildung 3.7A: Klassifikationsraten der Spektren nach Basislinienkorrektur mit robustem Polynomfit verschiedener Ordnung (1., 2., 4., 6., 8. Ordnung) sowie mit dem Whittaker Algorithmus. Die angezeigten Werte beziehen sich auf die Klassifikation mittels SVM. **Abbildung 3.7B**: Klassifikationsraten auf Basis der geschätzten Basislinien der Vorbehandlungsmethoden aus Abbildung 3.7A. Die angezeigten Werte sind wiederum die Klassifikationsraten der SVM.

Tabelle 3.4: Wiedererkennungsraten der verschiedenen Klassifikationsmethoden (PLS-DA, LDA, QDA, 3NN, SVM) auf Basis der „Spike"-eliminierten Spektren sowie auf Basis ausgewählter Wellenzahlen. Die Wellenzahlen der „Spike"-eliminierten Spektren, die nur Basisline (keine Peaks) enthalten, werden bei der Wellenzahlauswahl ausgeschlossen.

	PLS-DA	**LDA**	**QDA**	**MDA**	**3NN**	**SVM**
Ganzes Spektrum (SPIKEELIM)	71,7%	69.9%	78,9%	79,4%	83,5%	87,3%
Ausgewählte Wellenzahlen	70,1%	68,6%	78.5%	77,2%	80,8%	85,3%

3.4.1.2.2 Diskussion

Aus Abbildung 3.6 wird deutlich, dass sich die Basislinien je nach Grad des verwendeten Polynoms mehr oder weniger stark an die Raman-Spektren anpassen. Die Schätzung ist allerdings in allen Fällen so grob, dass der Einfluss von Peaks in den Basislinien visuell nicht erkennbar ist. Besonders beim Polynom 8. Ordnung sowie beim Whittaker-Algorithmus erhält man bei visueller Inspektion subjektiv betrachtet sehr gut geschätzte Basislinien und basislinienkorrigierte Spektren. Bei der Klassifikation stellt sich allerdings heraus, dass besonders bei höhergradigen Polynomen und beim Whittaker-Algorithmus die Klassifikationsrate der basislinienkorrigierten Spektren im Vergleich zu den „Spike"-eliminierten Rohspektren abnimmt. Außerdem erhält man überraschend hohe Wiedererkennungsraten, wenn nur die Basislinien klassifiziert werden (siehe Abbildung 3.7B). Dies kann man vor allem für SVMs feststellen. So werden bei der Klassifikation der Basislinien, die auf dem Whittaker-Algorithmus basieren, genauso viele Spektren richtig klassifiziert (77.9%) wie bei der Klassifikation der korrigierten Spektren selbst (77.5%). Je höher der Grad des Polynoms ist, desto schlechter wird die Klassifikationsrate aller Klassifizierer für die korrigierten Spektren und desto besser wird die Klassifikationsrate für die Basislinien. Je genauer sich die geschätzten Basislinien also an die tatsächliche Basislinie anpassen, desto mehr scheinen sie für die Klassifikation relevante Information zu beinhalten.

Grundsätzlich ist das Ergebnis nicht erstaunlich, da durch eine genauere Abbildung der Basislinie auch das Risiko steigt, dass fälschlicherweise spektrale Information in die Basislinienschätzung mit eingeht. Es ist jedoch unwahrscheinlich, dass die alleinige Ursache für den Verlust an Klassifikationsrate in der Verzerrung der spektralen Information liegt, da bereits bei der Klassifikation linearer Basislinien (BL-Poly1), die keine spektrale Information enthalten, eine Klassifikationsrate von 49,8% (SVM) erreicht wird. Die Basislinien scheinen also tatsächlich Informationen zu enthalten, die für eine Klassifikation genutzt werden können. Auch der zweite Test bestätigt diese Annahme (siehe Tabelle 3.4). Werden spektrale Teile entfernt, die ausschließlich Basislinie enthalten und wird mit diesen Spektren eine Klassifikation durchgeführt, erwartet man normalerweise, dass die Klassifikationsrate ansteigt, da Rauschen aus den Spektren entfernt wird. Nur wenn Information in der Basislinie vorhanden ist, ist mit einem Verlust an Klassifikationsrate zu rechnen. Der Vergleich der Klassifikationsraten unter Verwendung ausgewählter Wellenzahlen (850-1750 cm^{-1} und 2650-3150 cm^{-1}) mit den Ergebnissen der gesamten Spektren zeigt einen signifikanten Verlust an Klassifikationsrate durch die Entfernung der Basislinienanteile, was wiederum besonders für *k*NN und SVM zu beobachten ist. Die Abnahme der Klassifikationsrate ist nicht so stark wie bei der Basislinienkorrektur, da bei der Wellenzahlauswahl nur ein kleiner Spektralbereich entfernt wurde, während der größte Teil der Basislinie erhalten bleibt. Trotzdem ist das Ergebnis ein deutliches Indiz dafür, dass Klasseninformation in der Basislinie vorhanden ist. Diese Beobachtung kann verschiedene Ursachen haben. Zum einen kann die Information von den Mikroorganismen selbst stammen, was bedeuten würde, dass die charakteristische Fluoreszenz der Bakterien zur Differenzierung genutzt werden kann. Die Effekte können aber auch Artefakte sein (beispielsweise unterschiedliche Messbedingungen), die sich in den Spektren niederschlagen und nichts mit der Stammzugehörigkeit der Bakterien zu tun haben. Da nicht geklärt ist, woher die Klasseninformation der Basislinien stammt, ist es generell zuverlässiger, eine Klassifikation mit basislinienkorrigierten Spektren durchzuführen, selbst wenn dadurch leichte Einbußen in der Klassifikationsrate in Kauf genommen werden müssen. Diese Argumentation wird durch einige Studien unterstrichen, die zeigen, dass durch eine Basislinienkorrektur in der Regel robustere Ergebnisse hinsichtlich der Vorhersage neuer Objekte erzielt werden, als bei der Analyse nicht vorbehandelter Spektren [143].

In dieser Studie fällt auf, dass vor allem *k*NN und SVMs die zusätzliche Information aus der Basisline nutzen, um hohe Klassifikationsraten zu erzielen. So büßen diese Algorithmen ihre Überlegenheit gegenüber den anderen Klassifikationsmethoden durch eine Basislinienkorrektur weitgehend ein. Bei Anwendung des Whittaker-Algorithmus erhält man mit SVMs deshalb ähnliche Ergebnisse wie mit QDA und MDA. Für die parametrischen Methoden sind dagegen durch eine Basislinienkorrektur nur leichte Verluste in der Klassifikationsrate zu verzeichnen, die nicht signifikant sind. Dies wirft die Frage auf, ob die Information in der Basislinie grundsätzlich notwendig ist, um eine hohe Klassifikationsrate zu erzielen. Um dem nachzugehen, wurden zusätzlich zwei Methoden der paarweisen Klassifikation (PK-LDA und PK-MDA), deren Ergebnisse in einem späteren Kapitel der Arbeit (siehe Kapitel 3.4.3) beschrieben werden, herangezogen. Bei der paarweisen Klassifikation werden die klassischen Algorithmen (hier LDA und MDA) mit einer Binarisierung kombiniert (siehe Kapitel 2.3.3.5.1), was -wie in Kapitel 3.4.3.1 gezeigt wird- teilweise zu sehr hohen Klassifikationsraten führen kann. In Abbildung 3.8 ist dargestellt, wie sich die bisher besprochenen Klassifikationsmethoden und die paarweisen Versionen von LDA und MDA bei der Basislinienkorrektur verhalten.

Man beobachtet, dass PK-LDA und PK-MDA ähnlich stabil bei der Klassifikation der basislinienkorrigierten Spektren sind wie die bereits betrachteten parametrischen Methoden. Im Gegensatz dazu verlieren SVM und *k*NN massiv an Vorhersagegenauigkeit, was sich umso deutlicher zeigt, je höher der Grad des verwendeten Polynoms ist. PK-MDA, die bei den nicht basislinienkorrigierten Spektren eine vergleichbar gute Klassifikationsrate erreicht wie das SVM Modell, verliert deshalb beim Whittaker-Algorithmus nur leicht an Wiedererkennungsrate und erzielt dabei noch 83.1%. Die Klassifikationsrate der SVMs fällt dagegen auf 77.5% ab.

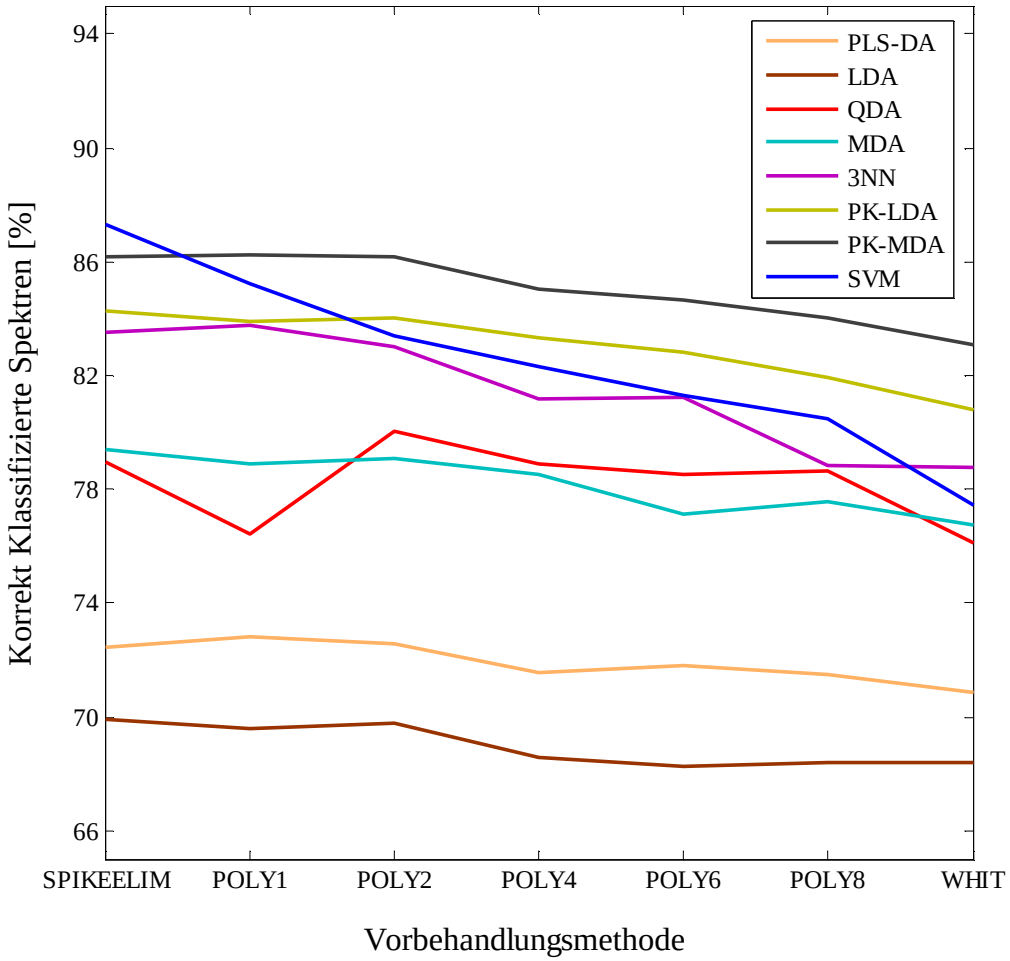


Abbildung 3.8: Klassifikationsraten verschiedener Klassifikationsmethoden mit Spektren nach Basislinienkorrektur durch robusten Polynomfit verschiedener Ordnung (1., 2., 4., 6., 8. Ordnung) sowie mit dem Whittaker Algorithmus.

Zusammenfassend kann man festhalten, dass die Basislinienkorrektur zu einer Verschlechterung der Klassifikationsrate bei allen Klassifikationsmethoden führt, wobei sich parametrische Klassifikationsmethoden stabiler als die nicht-parametrischen Methoden verhalten. Vor allem bei SVMs kommt es durch die Basisliniensubtraktion zu einem signifikanten Verlust an Vorhersagekraft. Es konnte gezeigt werden, dass in der Basislinie Klasseninformation steckt. Die Tatsache, dass SVMs die Basislinieninformation stärker nutzen als parametrische Klassifikationsmethoden deckt sich mit der Beobachtung, dass bei

SVMs auch durch eine Vektornormierung schlechtere Ergebnisse erhalten werden. Parametrische Methoden profitieren hingegen von der Vektornormierung. Da nicht geklärt ist, ob diese Informationen durch die Bakterien selbst oder durch Artefakte entstehen, liefert die Basislinienkorrektur möglicherweise robustere Modelle. Dies kann in dieser Arbeit nicht abschließend geklärt werden. Deshalb wurden die Vorbehandlungsmethoden gewählt, die die besten Klassifikationsraten zeigen (Vektornormierung für parametrische Methoden und „Spike"-eliminierte Rohspektren für nicht parametrische Methoden). Zukünftige Studien zur Vorhersage von Bakterienstämmen werden zeigen, ob sich diese Wahl als robust erweist, oder ob eine Basislinienkorrektur notwendig ist. Bei der Anwendung einer Basislinienkorrektur sind paarweise Ansätze der parametrischen Methoden (z. B. PK-MDA, PK-LDA) den SVMs überlegen. Ist für SVMs eine Basislinienkorrektur erforderlich, sollte der 1. Ableitung gegenüber den Methoden der Basisliniensubtraktion der Vorzug gegeben werden.

3.4.1.3 Kombination von Normierung und Basislinienkorrektur

Einige Studien über Datenvorbehandlung in der Raman-Spektroskopie biologischer Proben schlagen eine Kombination aus Vektornormierung und Basislinienkorrektur vor, um Störeffekte in den Raman-Spektren zu beseitigen [142]. Da in dieser Arbeit sowohl additive als auch multiplikative spektrale Effekte vorhanden sind, stellt dies eine naheliegende Variante der Datenvorbehandlung dar. Deshalb wird im Folgenden getestet, welchen Einfluss die Kombination aus Basislinienkorrektur und Vektornormierung auf das Klassifikationsergebnis hat. Dabei wird die Vektornormierung sowohl vor (N-POLY4, N-1.Abl, N-WHIT) als auch nach (POLY4-N, 1.Abl-N, WHIT-N) der Basislinienkorrektur durchgeführt. Anschließend werden die entstandenen Spektren klassifiziert. Die daraus erhaltenen Ergebnisse sind in Abbildung 3.9 dargestellt.

3.4.1.3.1 Ergebnisse

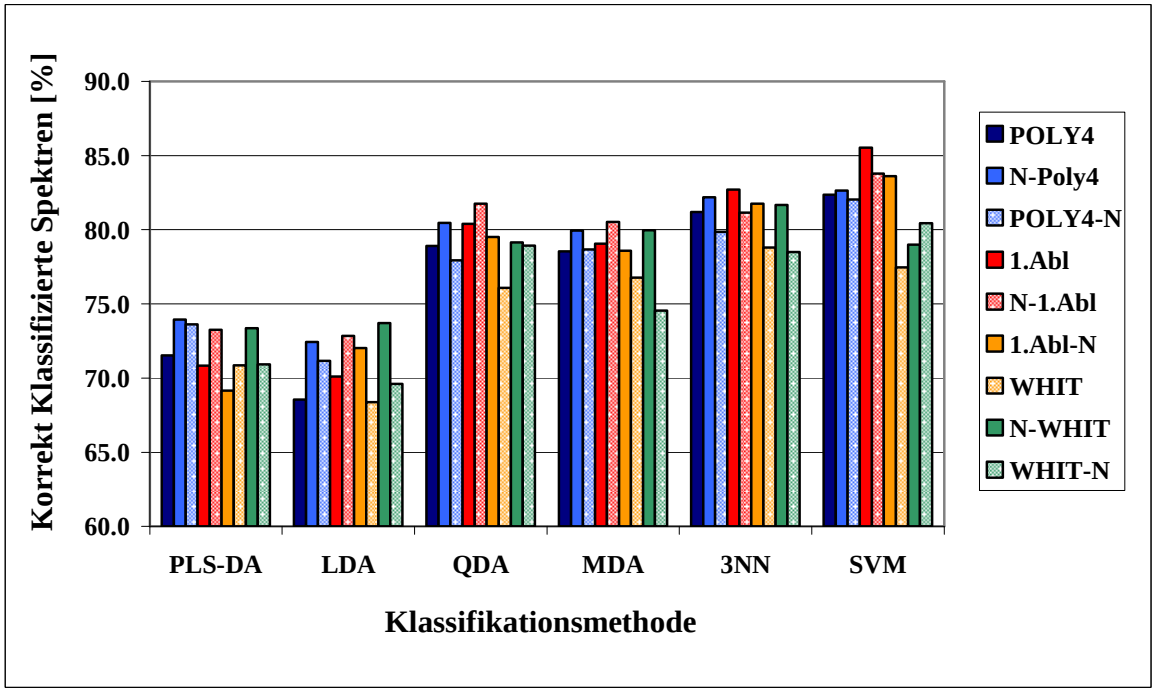


Abbildung 3.9: Klassifikationsraten bei verschiedenen Kombinationen aus Vektornormierung und Basisilinienkorrektur

3.4.1.3.2 Diskussion

Wie bei der Analyse der einzelnen Vorbehandlungsmethoden, so stellt man auch bei der Kombination von Vektornormierung und Basislinienkorrektur ein unterschiedliches Verhalten von parametrischen und nicht-parametrischen Klassifikationsmethoden fest (siehe Abbildung 3.9). Bei allen parametrischen Methoden steigt durch eine Normierung der Spektren vor der Basislinienkorrektur die Klassifikationsrate im Vergleich zur Verwendung der nur basislinienkorrigierten Spektren an. Die erhaltenen Ergebnisse sind vergleichbar mit den Ergebnissen einer einfachen Normierung. Bei einer Normierung nach der Basislinienkorrektur beobachtet man dagegen meist keine Verbesserung der Klassifikationsrate im Vergleich zu den basislinienkorrigierten Spektren. Auf diese Weise bestätigt sich die in Kapitel 3.4.1.1. aufgestellte Theorie, dass die Vektornormierung der

entscheidende Schritt bei der Vorbehandlung der Spektren für die parametrischen Methoden ist. Erfolgt die Normierung erst nach der Basislinienkorrektur, ist der positive Effekt der Normierung nicht mehr gegeben, da die Spektren bereits vor der Vektornormierung durch die Basislinienkorrektur stark verändert wurden. Die Normierung vor der Basislinienkorrektur wirkt sich dagegen positiv aus, da auf diese Weise die Normierung den Haupteinfluss auf die vorbehandelten Spektren hat. Wie in Kapitel 3.4.1.1 beschrieben, ist die Bildung der 1. Ableitung die einzige Methode der Basislinienkorrektur, die sich nicht negativ auf die Klassifikationsergebnisse der SVM auswirkt. Eine mögliche Begründung dafür ist, dass bei der 1. Ableitung im Vergleich zu den anderen Methoden wesentlich weniger Information aus den Spektren entfernt wird. So werden nur additive Effekte, die zu horizontal versetzten Basislinien führen, ausgeglichen. Ansonsten kommt es idealerweise zu keinem Informationsverlust bzw. zu keiner Verzerrung. Es kann allenfalls Rauschen verstärkt werden, was von der Art der Berechnung abhängt (siehe Kapitel 2.3.1.5a). Bei der Kombination von Vektornormierung und 1. Ableitung fällt die Leistungsfähigkeit des SVM Modells im Vergleich zur einfachen 1. Ableitung folglich wieder ab. Da sich sowohl die Vektornormierung als auch die Basisliniensubtraktion negativ auf das Klassifikationsergebnis der SVMs auswirken, erweist sich auch deren Kombination für die Klassifikation nicht als vorteilhaft.

3.4.2 Klassifikation

In den vorangehenden Kapiteln wurden die Auswirkungen verschiedener Datenvorbehandlungsmethoden auf die Klassifikation der Bakterienstämme analysiert. So wurde für die verwendeten Klassifikationsmethoden jeweils eine passende Vorbehandlungsmethode gefunden. Im Folgenden liegt der Fokus auf der Wahl einer geeigneten Klassifikationsmethode. Wie in Kapitel 2.3.3.2 beschrieben, hängt die Zweckmäßigkeit einer Klassifikationsmethode im hohen Maß von der Struktur der Daten sowohl innerhalb als auch zwischen den Klassen ab. Da die Datenstruktur im Vorfeld einer Klassifikation in der Regel nicht bekannt ist, muss dies durch Testen verschiedener Klassifikatonsalgorithmen herausgefunden werden. In dieser Arbeit wird ein hochdiverser Bakteriendatensatz (unterschiedliche Wachstumsbedingungen der Bakterien) analysiert. Aus

diesem Grund ist damit zu rechnen, dass innerhalb der einzelnen Klassen eine große Heterogenität besteht, was auf ein nichtlineares Klassifikationsproblem schließen lässt. Um dies zu überprüfen und um sich einen ersten Überblick über die Vorhersagegenauigkeit unterschiedlicher Methoden zu verschaffen, wurden zunächst verschiedene lineare (PLS-DA, LDA) sowie nichtlineare Methoden (QDA, MDA, *k*NN, SVMs) für die Klassifikation angewendet. Die Wiedererkennungsraten der Klassifikationsmethoden in Kombination mit verschiedenen Vorbehandlungsmethoden sind aus Abbildung 3.3 zu entnehmen. Abbildung 3.10 zeigt die Haupteffekte der verschiedenen Methoden nach Durchführung einer zweifaktoriellen CVANOVA mit anschließendem Test nach Tukey.

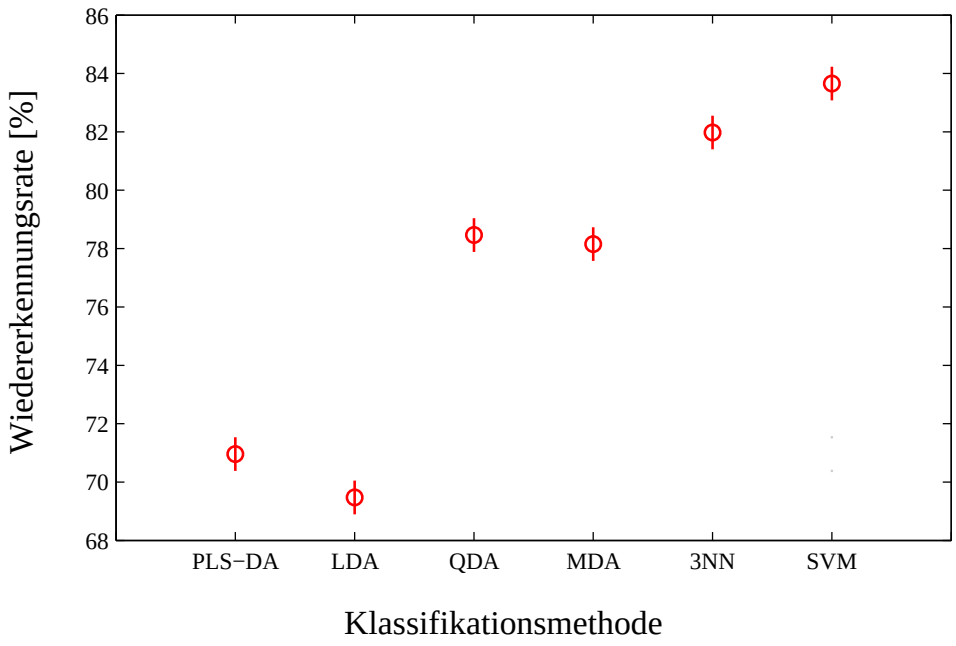


Abbildung 3.10: Im Anschluss an eine signifikante zweifaktorielle CVANOVA, wurde Tukey´s Test verwendet, um signifikante Unterscheide (Haupteffekte) zwischen den Wiedererkennungsraten der verschiedenen Klassifikationsmethoden zu finden. Die Kreise kennzeichnen die durchschnittliche Wiedererkennungsrate der verschiedenen Klassifikationsmethoden. Alle Vorbehandlungsmethoden werden dabei gleichzeitig berücksichtigt. Die Balken beschreiben die Konfidenzintervalle. Überlappende Konfidenzintervalle zeigen an, dass die Klassifikationsraten sich nicht signifikant unterscheiden ($\alpha < 0.05$).

Es fällt auf, dass nichtlineare Klassifikationstechniken (SVMs, 3NN, QDA und MDA) lineare Klassifikatoren (LDA und PLS-DA) signifikant in der Vorhersagegenauigkeit übertreffen. Unter den nichtlinearen Methoden zeichnen sich vor allem die nicht-parametrischen Methoden SVMs und 3NN durch eine hohe Leistungsfähigkeit aus. Sie erreichen signifikant höhere Klassifikationsraten als die parametrischen, nichtlinearen Techniken (QDA, MDA), die untereinander sehr ähnliche Ergebnisse erzielen. Die beste Klassifikationsrate wird mit Hilfe von SVMs (87.3% richtig klassifizierte Spektren) erreicht (siehe Abbildung 3.3). SVMs erweisen sich in der Klassifikation somit als signifikant besser als alle anderen Klassifikationsmethoden (siehe Abbildung 3.10). Diese Ergebnisse machen deutlich, dass flexible Entscheidungsgrenzen die Vorhersagegenauigkeit für das vorliegende Klassifikationsproblem erhöhen.

Die Beobachtung, dass vor allem nichtlineare Methoden bei der Klassifikation des Datensatzes gut abschneiden, lassen darauf schließen, dass für parametrische Klassifikationsmethoden, die sich durch zahlreiche attraktive Eigenschaften (z. B. Einfachheit in der Durchführung, leichte Interpretierbarkeit usw.) auszeichnen, die Klassifikationsergebnisse verbessern lassen, indem ein gewisser Grad an Nichtlinearität eingeführt wird. Eine höhere Nichtlinearität kann beispielsweise durch die in Kapitel 2.3.3.5 beschriebene paarweise Klassifikation erreicht werden. Im folgenden Kapitel (Kapitel 3.4.3) werden die Auswirkungen der paarweisen Klassifikation auf die Leistungsfähigkeit aller hier verwendeten Klassifikatoren diskutiert. Eine Ausnahme sind SVMs, bei denen sowieso eine paarweise Klassifikation stattfindet.

3.4.3 Paarweise Klassifikation

Durch paarweise Klassifikation (siehe Kapitel 2.3.3.5) können lineare Klassifikationsmethoden wie LDA nichtlineare Entscheidungsgrenzen bilden. Daneben kann bei Klassifikationsmethoden, die sich bereits durch nichtlineare Entscheidungsgrenzen auszeichnen (MDA), die Flexibilität der Entscheidungsgrenzen durch eine Binarisierung zusätzlich gesteigert werden. Da für die Analyse des diversen bakteriellen Datensatzes nichtlineare Klassifikatoren besser geeignet sind als lineare Methoden (siehe Kapitel 3.4.2),

wurde versucht, durch die paarweise Klassifikation („One-Against-One“ Methode) den Klassifikationserfolg weiter zu verbessern. Dabei wurden zwei Ansätze verwendet. Zunächst kam das einfache „Major Voting“ Verfahren zum Einsatz. Anschließend wurde die Leistungsfähigkeit verschiedener Methoden zur Bildung von Multi-Klassen *a posteriori* Wahrscheinlichkeiten überprüft.

3.4.3.1 “Major Voting”

Das Multi-Klassen-Problem wurde zunächst in mehrere Zwei-Klassen-Probleme gemäß dem „One-Against-One“–Verfahren aufgeteilt (siehe Kapitel 2.3.3.5.1). Die daraus erhaltenen Zwei-Klassen-Entscheidungen wurden anschließend mit Hilfe des “Major Vote”-Verfahrens (siehe Kapitel 2.3.3.5.1) zu einer Multi-Klassen-Entscheidung umgerechnet. Die Gegenüberstellung der Ergebnisse von einfacher und paarweiser Klassifikation („One-Against-One“, „Major Voting“) sind in Tabelle 3.5 aufgeführt.
Zur Beurteilung der Ergebnisse wurden die Unterschiede zwischen den paarweisen Klassifikationsmethoden mit einer einfaktoriellen CVANOVA und anschließendem Tukey´s Test analysiert, was in Abbildung 3.11 dargestellt ist.

Tabelle 3.5: Wiedererkennungsraten (%) der untersuchten Klassifikationsmethoden durch einfache und paarweise Klassifikation (50-fache Kreuzvalidierung). Die für die Klassifikation verwendeten Parameter sind folgende: PLS-DA: 10 Faktoren LDA: 30 PCs; QDA: 15 PCs; MDA: 4 Subzentren, 30 PCs; 3NN = *k*NN: 20 PCs, *k* = 3; SVM: RBF-Kernel: $C=1x10^{5}$, $\gamma=5x10^{-3}$.

	PLS-DA	**LDA**	**QDA**	**MDA**	**3NN**	**SVM**
Einfache Klassifikation	73.8%	73.6%	80.9%	80.9%	83.5%	
Paarweise Klassifikation	82.5%↑	84.1%↑	80.8%↔	86.6%↑	83.5%↔	87.3%

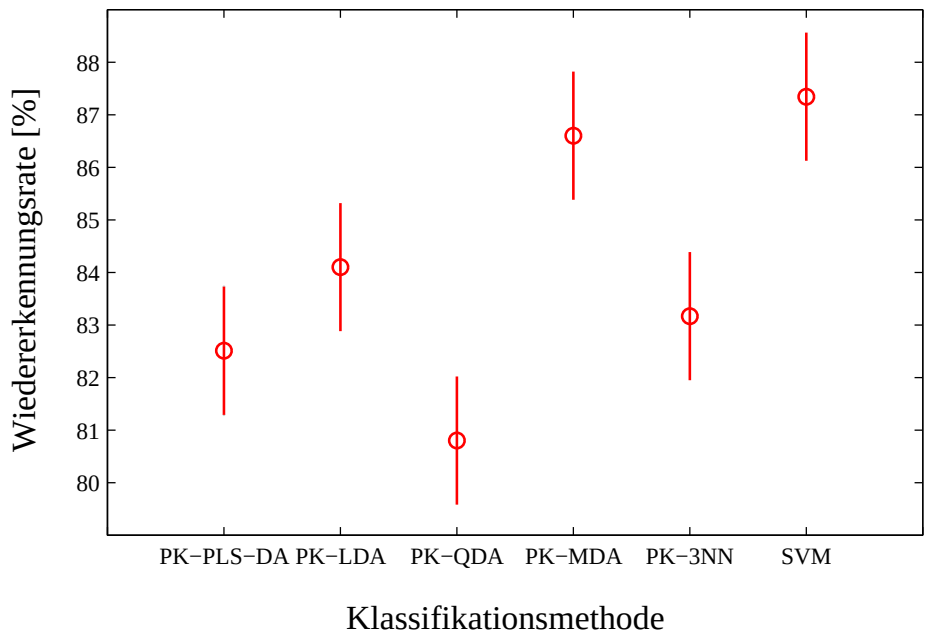


Abbildung 3.11: Eine einfaktorielle CVANOVA ließ auf signifikante Unterschiede zwischen den paarweisen Klassifikationstechniken schließen. Anschließend wurde Tukey´s Test durchgeführt, um statistisch signifikante Unterschiede zwischen den Klassifikationsraten der paarweisen Klassifikationsmethoden zu finden. Die Balken beschreiben die Konfidenzintervalle. Überlappende Konfidenzintervalle zeigen an, dass die Klassifikationsraten sich nicht signifikant unterscheiden ($\alpha < 0.05$).

Aus Tabelle 3.5 und Abbildung 3.11 geht hervor, dass sich die Vorhersagegenauigkeit der linearen Klassifikationsmethoden (PLS-DA, LDA) sowie der MDA durch die paarweise Klassifikation signifikant erhöht. Auf diese Weise erreichen PLS-DA und LDA ebenso gute Klassifikationsraten wie die nichtlinearen Methoden QDA und 3NN, für die sich die Klassifikationsraten nicht wesentlich verändern. Es sei angemerkt, dass SVMs den paarweisen Ansatz sowieso verwenden, um Multi-Klassen-Probleme zu lösen. Der große Gewinn an Vorhersagekraft für PLS-DA und LDA kann auf die Nichtlinearität zurückgeführt werden, die durch die Binarisierung in das Modell eingeführt wird. Auch die Klassifikation mittels MDA, die bereits nichtlineare Entscheidungsgrenzen bilden kann, wird durch die

Binarisierung wesentlich verbessert. Dies bewirkt die bedeutendste Steigerung an Vorhersagekraft. So steigt die Wiedererkennungsrate auf 86.6% an, was sich vom Ergebnis der SVMs (87,5%) statistisch nicht signifikant unterscheidet (siehe Abbildung 3.11).

3.4.3.2 Bildung von Multi-Klassen-Wahrscheinlichkeiten

Für die Methoden, die in Kapitel 3.4.3.1 die besten Vorhersagewerte aufweisen (LDA, MDA, SVM), wurden neben dem einfachen „Major Voting"-Verfahren verschiedene Techniken zur Bildung von Multi-Klassen-Wahrscheinlichkeiten (HT, PKPD, Wu) [85-87] getestet. Dabei wurde untersucht, ob sich durch das Einbeziehen der *a posteriori* Wahrscheinlichkeiten der Zwei-Klassen-Entscheidungen in die Berechnung der Multi-Klassen-Zugehörigkeiten eine im Vergleich zum einfachen „Major Voting" verbesserte Vorhersagekraft erreichen lässt. Daneben war es von Interesse, ob Unterschiede zwischen den Methoden zur Bildung von *a posteriori* Wahrscheinlichkeiten für die Multi-Klassen-Zuordnung (HT, PKPD,Wu) festzustellen sind. Die erhaltenen Wiedererkennungsraten dieser Techniken sind in Tabelle 3.6 zusammengefasst.

Tabelle 3.6: Klassifikationsraten (%) von paarweiser LDA, paarweiser MDA und SVM unter Verwendung von „Major Voting" sowie verschiedener Methoden zur Bildung von Muli-Klassen *a posteriori* Wahrscheinlichkeiten nach Binarisierung. Die berücksichtigten Methoden sind der „Pairwise Coupling" Algorithmus nach Hastie und Tibshirani (HT) [85], der Algorithmus nach nach Price, Knerr, Personnaz und Dreyfus (PKPD) [87] sowie der Algorithmus nach Wu et al. [86].

	LDA	**MDA**	**SVM**
Major Voting	84.1%	86.6%	87.3%
HT	84.4%	85.9%	83.4%
PKPD	84.8%	86.5%	85.8%
Wu	84.5%	86.0%	86.2%

Die Ergebnisse in Tabelle 3.6 lassen erkennen, dass durch keine der Methoden zur Berechnung der Multi-Klassen *a posteriori* Wahrscheinlichkeiten die Klassifikationsraten signifikant verbessert werden können. Es ist aber auch keine Verschlechterung zu verzeichnen mit Ausnahme des „Pairwise Coupling"–Algorithmus von Hastie und Tibshirani (HT), der bei SVMs zu einem signifikanten Verlust an Vorhersagegenauigkeit führt. In Kombination mit PK-LDA und PK-MDA erweist sich dieser Algorithmus dagegen als stabil.
Daraus lässt sich folgern, dass für die Klassifikation der Bakterien das einfache „Major Voting" Verfahren ausreicht, um eine gute Vorhersagegenauigkeit zu erreichen. Eine Kombination der Klassifikationsmethoden mit den Verfahren der Bildung von Multi-Klassen *a posteriori* Wahrscheinlichkeiten ist nur dann erforderlich, wenn *a posteriori* Wahrscheinlichkeiten geschätzt werden müssen, was sich häufig als nützlich erweist. In diesem Fall sind alle drei Methoden für die Schätzung der *a posteriori* Wahrscheinlichkeiten geeignet, ohne dass es zu einem Verlust an Vorhersagegenauigkeit kommt. Eine Ausnahme bilden SVMs, bei denen die Algorithmen nach Price, Knerr, Personnaz und Dreyfus (PKPD) und Wu et al. (Wu) dem Algorithmus nach Hastie und Tibshirani (HT) vorzuziehen sind.
Die zwei Klassifikationsmethoden, die sich für die Raman-spektroskopische Differenzierung von einzelnen Bakterienzellen am besten eignen, sind unter den hier verglichenen Techniken SVMs und paarweise MDA. Diese beiden Methoden werden im weiteren Verlauf der Arbeit bezüglich ihrer Eigenschaften für die gegebene Aufgabenstellung charakterisiert. Dabei wird die Zuverlässigkeit der Methoden nach der Modellselektion sowie deren Robustheit gegenüber kleineren Trainingsdatensatzgrößen überprüft. Zudem wird die Interpretierbarkeit des MDA-Modells demonstriert, womit Zusammenhänge zwischen Wachstumsbedingungen, Datenstruktur und Klassifikationserfolg dargestellt werden können. Abschließend werden die Methoden dazu verwendet, um unbekannte, nicht im Trainingsdatensatz vorhandene Testobjekte als Vorhersageausreißer zu erkennen.
Zunächst werden jedoch die Klassifikationsergebnisse der beiden Methoden nocheinmal genauer betrachtet. Dabei ist es vor allem von Interesse wie sich die Klassifikationsrate in Bezug auf die einzelnen Stämme verhält.

Es stellen sich folgende Fragen:

1. Lassen sich einige Bakterienstämme besonders leicht oder besonders schwer differenzieren und gibt es dabei eine Abhängigkeit von den verwendeten Klassifikationsmethoden?
2. Welche Klassifikationsrate erhält man auf Artebene?
3. Hängt die Klassifikationsrate mit der Anzahl der verwendeten Spektren pro Stamm zusammen? Eine deutliche Korrelation würde darauf hinweisen, dass für manche Bakterienstämme zu wenig Spektren aufgenommen wurden. In diesem Fall müsste der Trainingsdatensatz vergrößert werden.

Um Antworten auf diese Fragen zu erhalten, sind in Tabelle 3.7 die Klassifikationsraten der einzelnen Bakterienstämme auf Stamm- und Artebene aufgelistet. Spearman´s Korrelationskoeffizient ρ ist eine geeignete Maßzahl zur Prüfung auf positive Korrelationen zwischen den Wiedererkennungsraten und der Anzahl an Spektren pro Stamm [48]. Die Berechnung von ρ zeigt keine positive Korrelation ($\rho_{\text{paarweise MDA}} = -0.3218; \rho_{\text{SVM}} = -0.0875$). Fehlklassifikationen können also nicht in erster Linie auf eine zu kleine Trainingsdatenmenge zurückgeführt werden. Aus Tabelle 3.7 geht außerdem hervor, dass die besten Klassifikationsraten bei den Stämmen der Arten *M. luteus*, *M. lylae* und *S. cohnii* zu verzeichnen sind, während es schwer ist, die Stämme der Arten *S. epidermidis* und *E. coli* zu differenzieren. Dies gilt sowohl für paarweise MDA als auch für SVMs. Fehlklassifikationen kommen dabei vor allem innerhalb der Arten vor. Verwechslungen zwischen den verschiedenen Arten sind selten. So ist der Prozentsatz korrekt klassifizierter Spektren auf Artebene für paarweise MDA 97.4%, was einer annähernd perfekten Differenzierung auf Artebene entspricht und für das „Online-Monitoring“ im industriellen Umfeld gut geeignet ist.

Tabelle 3.7: Klassifikationsraten (%) der untersuchten Reinraumbakterien auf Stamm- und Artebene. Die berücksichtigten Klassifikationsmethoden sind paarweise MDA und SVMs unter Verwendung der 50-fachen Kreuzvalidierung.

Name	Anzahl der Spektren	Stammebene		Artebene	
		Paarweise MDA	SVM	Paarweise MDA	SVM
B. pumilus DSM 27	57	84.2	78.9	93.0	86.0
B. pumilus DSM 361	69	73.9	84.1	82.6	92.8
B. sphaericus DSM 28	53	86.8	86.8	90.6	92.5
B. sphaericus DSM 396	42	76.2	85.7	76.2	85.7
B. subtilis DSM 10	326	95.1	96.9	95.1	97.2
B. subtilis DSM 347	42	90.5	89.6	97.6	91.7
M. luteus DSM 20030	48	93.8	99.8	95.8	99.8
M. luteus DSM 348	619	99.8	97.8	99.8	97.8
M. lylae DSM 20315	45	97.8	95.0	97.8	95.0
M. lylae DSM 20318	20	100.0	96.9	100.0	98.4
S. cohnii DSM 20260	64	98.4	95.2	98.4	95.2
S. cohnii DSM 6669	62	100.0	85.2	100.0	88.5
S. cohnii DSM 6718	61	80.3	83.6	91.8	95.1
S. cohnii DSM 6719	61	77.0	63.4	91.8	99.1
S. epidermidis DSM 1798	112	64.3	95.9	94.6	95.9
S. epidermidis DSM 44195	74	90.5	59.4	97.3	100.0
S. epidermidis DSM 20042	106	63.2	53.8	99.1	100.0
S. epidermidis DSM 3269	93	50.5	60.0	100.0	100.0
S. epidermidis DSM 3270	110	66.4	99.4	97.3	99.4
S. epidermidis ATCC 35984	805	99.1	95.4	99.4	98.5
S. warneri DSM 20036	65	93.8	83.6	96.9	91.0
S. warneri DSM 20316	67	82.1	85.7	88.1	95.2
E. coli DSM 1058	68	86.8	77.9	100.0	100.0
E. coli DSM 2769	108	62.0	79.6	99.1	100.0
E. coli DSM 423	112	80.4	70.5	99.1	92.0
E. coli DSM 429	90	64.4	63.3	100.0	100.0
E. coli DSM 498	86	70.9	72.1	100.0	100.0
E. coli DSM 499	83	67.5	50.6	100.0	98.8
E. coli DSM 613	94	53.2	83.0	98.9	100.0
Mittelwert (%)		**81.0**	**81.7**	**95.9**	**96.1**
Prozentsatz korrekt klassifizierter Spektren (Klassifikationsrate)	**3642**	**86.6**	**87.3**	**97.4**	**97.2**

3.4.4 Einfluss von Parameteroptimierung auf die Klassifikation

3.4.4.1 "Overfitting" durch Parameteroptimierung

Im bisherigen Verlauf der Studie fanden die Auswahl der „Tuning"-Parameter und die Bestimmung der Klassifikationsrate in einem Schritt statt. Dazu wurde eine einfache 50-fache Kreuzvalidierung durchgeführt. Um den Grad des „Overfittings" durch Modellselektion einschätzen zu können, ist es jedoch unverzichtbar, von der Modellselektion unabhängige Testdaten vorherzusagen, um auf dieser Basis die Modellgüte abzuschätzen (siehe Kapitel 2.3.4.1). Deshalb wurde für die zwei Klassifikationsmethoden, die in der 50-fachen Kreuzvalidierung die besten Klassifikationsergebnisse erzielten (PK-MDA und SVM), ein doppeltes Validierungsschema (siehe Abbildung 2.19) herangezogen. Dabei wurden eine innere und eine äußere Kreuzvalidierungsschleife durchlaufen, was die Bestimmung der Klassifikationsrate unabhängig von dem Modellselektionsprozess erlaubt. Bei der paarweisen MDA wurden zur Modellselektion in der inneren Schleife für alle Kombinationen aus PCs (5, 10, 15, ..., 60) und Subzentren (2, 4, 6, 8) die Klassifikationsraten berechnet. Die Parameterkombination mit der besten Klassifikationsrate in der inneren Schleife wurde anschließend zur Vorhersage des externen Testsets verwendet. Analog dazu wurde für SVMs in der inneren Schleife ein "Grid-search" durchgeführt. Dabei wurden für die zwei zu bestimmenden „Tuning"-Parameter in einem zuvor definierten Intervall mehrere Werte so festgelegt, dass das gesamte Intervall abgedeckt ist. Anschließend wurde mit allen Parameterkombinationen die innere Kreuzvalidierungsschleife durchlaufen. Die Parameterkombination, die dabei die beste Klassifikationsrate erzielte, diente jeweils zur Vorhersage des externen Testsets. In dieser Arbeit erfolgte für SVMs eine grobe (engl. Rough Tuning) sowie eine feine Suche (engl. Fine Tuning). Bei der groben Suche wurde für beide Parameter ein relativ großes Intervall abgetastet. In dem Bereich, in dem bei der groben Suche die besten Klassifikationsraten erzielt wurden, erfolgte anschließend eine genauere, engmaschigere Überprüfung der Parameterwerte. Da die Werte von C und γ zwischen 0 und Unendlich liegen, erfolgte die grobe Suche im Bereich von $2 \cdot 10^{-5}$ bis $2 \cdot 10^{10}$ (C und γ = $2 \cdot 10^{-5}$, $2 \cdot 10^{-4}$, $2 \cdot 10^{-3}$, …, $2 \cdot 10^{8}$, $2 \cdot 10^{9}$, $2 \cdot 10^{10}$). Für die anschließende feine Suche wurden

folgende Parameter verwendet: $C = 5 \cdot 10^4$, $6 \cdot 10^4$, …, $6 \cdot 10^5$, $7 \cdot 10^5$ und $\gamma = 5 \cdot 10^{-4}$, $6 \cdot 10^{-4}$, …, $6 \cdot 10^{-3}$, $7 \cdot 10^{-3}$.

3.4.4.1.1 Ergebnisse

Die Parameterkombination, die in der inneren Kreuzvalidierungsschleife der PK-MDA am häufigsten gewählt wurde, ist 8 Subzentren und 30 PCs. Außerdem sind 4 und 6 Subzentren mehrmals in der besten Parameterkombination enthalten. Für SVMs wurde in allen 50 Durchläufen der groben Suche die gleiche Parameterkombination gewählt ($C = 2 \cdot 10^5$, $\gamma = 2 \cdot 10^{-3}$). Bei der feinen Suche variierten dagegen die Werte im Bereich von $C \in \{5 \cdot 10^4; 4 \cdot 10^5\}$ und $\gamma \in \{8 \cdot 10^{-4}; 7 \cdot 10^{-3}\}$. In Tabelle 3.8 sind die durchschnittlichen Klassifikationsraten und die Standardabweichung der Vorhersagen der äußeren Kreuzvalidierungsschleife notiert.

Tabelle 3.8: Externe Validierungsergebnisse von PK-MDA und SVM (RBF-Kernel) bei der Verwendung von 2 Schleifen der 50-fachen Kreuzvalidierung

	Intervall der gewählten Parameter	**Klassifikationsrate**	**Standard-abweichung**
PK-MDA	Anzahl an Subzentren ϵ {4 ; 8} Anzahl an PCs ϵ {25 ; 40}	86.3%	3.8%
SVM – RBF Kernel (grobe Suche)	$C = 2 \cdot 10^5$ $\gamma = 2 \cdot 10^{-3}$	87.0%	3.5%
SVM – RBF Kernel (feine Suche)	$C \epsilon \{5 \cdot 10^4 ; 4 \cdot 10^5\}$ $\gamma \epsilon \{8 \cdot 10^{-4} ; 7 \cdot 10^{-3}\}$	87.0%	3.6%

3.4.4.1.2 Diskussion

Bei der Durchführung des doppelten Validierungsschemas (2 Schleifen einer 50-fachen Kreuzvalidierung) beobachtet man, dass sich die externen Validierungsergebnisse nicht signifikant von den Klassifikationsergebnissen, die bei der einfachen 50-fachen Kreuzvalidierung erhalten werden, unterscheiden (siehe Tabelle 3.8). Paarweise MDA erzielt 86.6% richtig klassifizierter Spektren unter der Verwendung der einfachen Kreuzvalidierung (30 PCs, 4 Subzentren) und 86.3% mit dem doppelten Validierungsschema. Die SVMs erreichen eine Klassifikationsrate von 87.3% bei der einfachen Kreuzvalidierung (C=$1 \cdot 10^5$, γ=$5 \cdot 10^{-3}$) und 87.0% unter Verwendung des doppelten Validierungsschemas (siehe Tabelle 3.8, feine Suche). Es stellte sich heraus, dass für SVMs die feine Suche (87.0%) keine besseren Ergebnisse liefert als die grobe Suche (87.0%). Abbildung 3.12 zeigt den "Contour-Plot" (Abbildung der Klassifikationsraten verschiedener Parameterkombinationen bei 50-facher Kreuzvalidierung) für paarweise MDA und SVM. Man erkennt, dass die Klassifikationsraten von paarweiser MDA im Intervall von 20 bis 45 PCs annähernd konstant bleiben. Dabei schneiden 4, 6, und 8 Subzentren besser ab als 2 Subzentren. Die Ergebnisse der SVM sind in dem Bereich $C \in \{2 \cdot 10^4; 2 \cdot 10^{10}\}$ und $\gamma \in \{2 \cdot 10^{-1}; 2 \cdot 10^{-4}\}$ nahezu konstant und zeigen ein leichtes lokales Maximum bei $C = 2 \cdot 10^5$ und $\gamma = 2 \cdot 10^{-3}$.

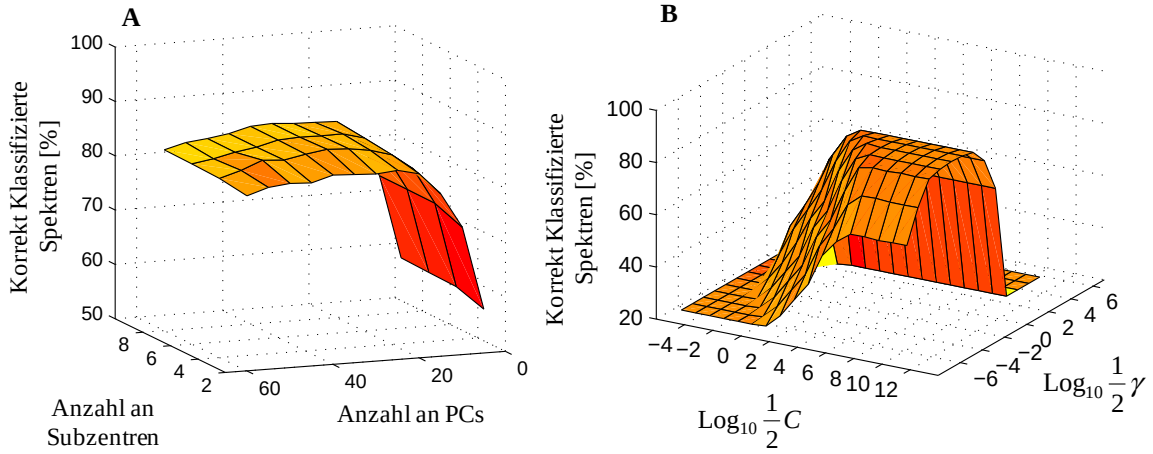


Abbildung 3.12A: „Contour-Plot" (Abbildung der Klassifikationsraten verschiedener Parameterkombinationen bei 50-facher Kreuzvalidierung) für paarweise MDA. **Abbildung 3.12 B**: „Contour-Plot" für SVMs (grobe Suche).

Der gleichmäßige Verlauf in großen Teilen beider Abbildungen deutet darauf hin, dass sich die Klassifikationsraten bei variierenden Parametern nur leicht ändern. Aus diesem Grund kann das Risiko des „Overfittings“ durch Modellselektion für beide Klassifikationsmethoden als gering eingestuft werden. Diese Annahme wird dadurch bestätigt, dass bei der einfachen und bei der doppelten Validierung sehr ähnliche Klassifikationsraten beobachtet werden. Daraus kann man schließen, dass eine einfache 50-fache Kreuzvalidierung in dieser Studie bereits eine realistische Einschätzung der Klassifikationsrate erlaubt.

3.4.4.2 Robustheit

Um die Robustheit der Modelle gegenüber kleineren sowie diverseren Trainingsdatensätzen einschätzen zu können, wurde das zweifache Validierungsschema, das bereits in Kapitel 3.4.4.1 zur Anwendung kam, nochmals mit „Bootstrapping“ in der der äußeren Schleife und 50-facher Kreuzvalidierung in der inneren Schleife durchgeführt. Bei dem „Bootstrap“-Verfahren landen in jedem Validierungsdurchlauf ungefähr 37% der Daten im Testset, während bei der 50-fachen Kreuzvalidierung jeweils nur 2% der Daten auf das Testset entfallen. Durch das Ziehen mit Zurücklegen führt „Bootstrapping“ außerdem zu einer deutlich stärkeren Durchmischung der Daten, die für die Parameterauswahl in der inneren Schleife sowie für die Testdatenvorhersage in der äußeren Schleife verwendet werden. Das Ziehen der „Bootstrap“-Stichproben wurde in dieser Arbeit für jede der Klassen getrennt durchgeführt, so dass ungefähr der gleiche Prozentsatz an Objekten aus jeder Klasse in das Testset eingeht. Um die optimale Anzahl an „Bootstrap“-Läufen zu ermitteln, wurde im Vorfeld der doppelten Validierung ein einfaches „Bootstrapping“ mit festgelegten Parametern und unterschiedlicher Anzahl an gezogenen „Bootstrap“-Stichproben durchgeführt (siehe Abbildung 3.13).

3.4.4.2.1 Ergebnisse

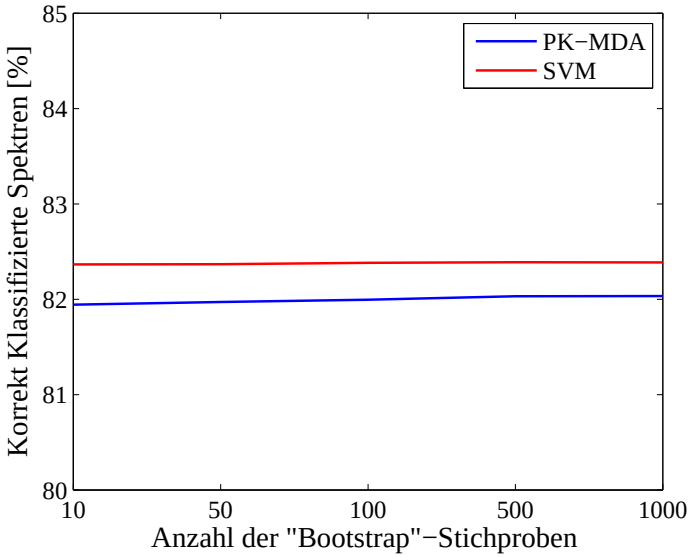


Abbildung 3.13: Klassifikationsraten von SVMs (RBF-Kernel, $C=1 \cdot 10^5$, $\gamma=5 \cdot 10^{-3}$) und paarweiser MDA (4 Subzentren, 30 PCs) bei verschiedener Anzahl an „Bootstrap“-Stichproben (10, 50, 100, 500 und 1000 Stichproben).

Aus Abbildung 3.13 ist ersichtlich, dass die Vorhersagegenauigkeit für eine unterschiedliche Anzahl an „Bootstrap“-Läufen annähernd konstant ist. Die Klassifikationsraten (Mittlerer Prozentsatz korrekt klassifizierter Testobjekte über alle „Bootstrap“-Stichproben) von paarweiser MDA liegen zwischen 83.9% (10 „Bootstrap“-Stichproben) und 84.1% (1000 „Bootstrap“-Stichproben). Die Werte der SVMs liegen zwischen 84.7% (10 „Bootstrap“-Stichproben) und 84.8% (1000 „Bootstrap“-Stichproben). 10 „Bootstrap“-Stichproben liefern also annähernd das gleiche Ergebnis wie 1000 „Bootstrap“-Stichproben. Im Folgenden wurden 100 „Bootstrap“-Stichproben für die Validierung verwendet. Das Ergebnis der doppelten Validierung mit „Bootstrapping“ in der äußeren Schleife ist in Tabelle 3.9 gezeigt.

Tabelle 3.9: Externe Validierungsergebnisse von paarweiser MDA und SVM (RBF-Kernel) mit 2 Validierungsschleifen. „Bootstrapping“ wurde in der äußeren Schleife verwendet, während in der inneren Schleife eine 50-fache Kreuzvalidierung ausgeführt wurde.

	Intervall der gewählten Parameter	**Klassifikations-rate**	**Standard-abweichung**
PK-MDA	Anzahl der Subzentren ϵ {6 ; 8} Anzahl an PCs ϵ {25 ; 45}	83.7%	1.2%
SVM – RBF Kernel (grobe Suche)	C ϵ $\{2 \cdot 10^5 ; 2 \cdot 10^7\}$ γ ϵ $\{2 \cdot 10^{-2} ; 2 \cdot 10^{-4}\}$	84.7%	0.9%
SVM – RBF Kernel (feine Suche)	C ϵ $\{6 \cdot 10^4 ; 6 \cdot 10^6\}$ γ ϵ $\{2 \cdot 10^{-4} ; 7 \cdot 10^{-3}\}$	84.7%	0.8%

3.4.4.2.2 Diskussion

Es stellte sich heraus, dass durch das „Bootstrapping“ in der äußeren Schleife die Klassifikationsraten beider Techniken (paarweise MDA und SVMs) im Vergleich zu den zwei Schleifen der 50-fachen Kreuzvalidierung um 2-3% Prozent abnehmen (siehe Tabelle 3.8 und Tabelle 3.9). Da die Klassifikationrate generell mit kleiner werdenden Trainingsdatensätzen schlechter wird, kann der Verlust an Vorhersagegenauigkeit auf den reduzierten Trainingsdatensatz, der sich mit dem neuen Schema der Stichprobenziehung ergibt, zurückgeführt werden. Die externen Validierungsergebnisse unterscheiden sich auch in diesem Fall nicht signifikant von den Klassifikationsraten, die bei der Durchführung eines einfachen „Bootstrappings“ für die Parameterauswahl erhalten werden. Dadurch wird das Ergebnis aus Kapitel 3.4.4.1 bestätigt, nach dem das Risiko des "Overfittings" durch die Parameteroptimierung (engl. Model Selection Bias) vernachlässigbar gering ist. So erreichte die paarweise MDA 84.5% an korrekt klassifizierten Spektren unter Verwendung des einfachen „Bootstrappings“ (8 Subzentren, 30 PCs) und 83.7% bei der Durchführung der zwei Schleifen des „Resamplings“ (siehe Tabelle 3.9). Die SVMs erzielten 84.8% mit einfachem „Bootstrapping“ ($C=1 \cdot 10^5$, $\gamma=5 \cdot 10^{-3}$) und 84.7% bei dem doppelten

Validierungsschema (siehe Tabelle 3.9). Abbildung 3.14 zeigt die „Contour-Plots“ für paarweise MDA und SVM unter Verwendung einer einfachen „Bootstrap“-Validierung. Dabei erkennt man keine Unterschiede zu den „Contour-Plots“ der 50-fachen Kreuzvalidierung (siehe Abbildung 3.12). Über große Teile erhält man einen gleichmäßigen Verlauf, was darauf hinweist, dass der Modellselektionsschritt hier unkritisch ist.
Neben der leichten Abnahme der Vorhersagegenauigkeit bei Verwendung des „Bootstrap“-Verfahrens in der äußeren Validierungsschleife, werden die Intervalle und die Varianz der in der inneren Schleife gewählten Paramter größer (siehe Tabelle 3.8 und Tabelle 3.9). Dies spiegelt die höhere Diversität der Daten, die in der inneren Schleife verwendet wurden wider.

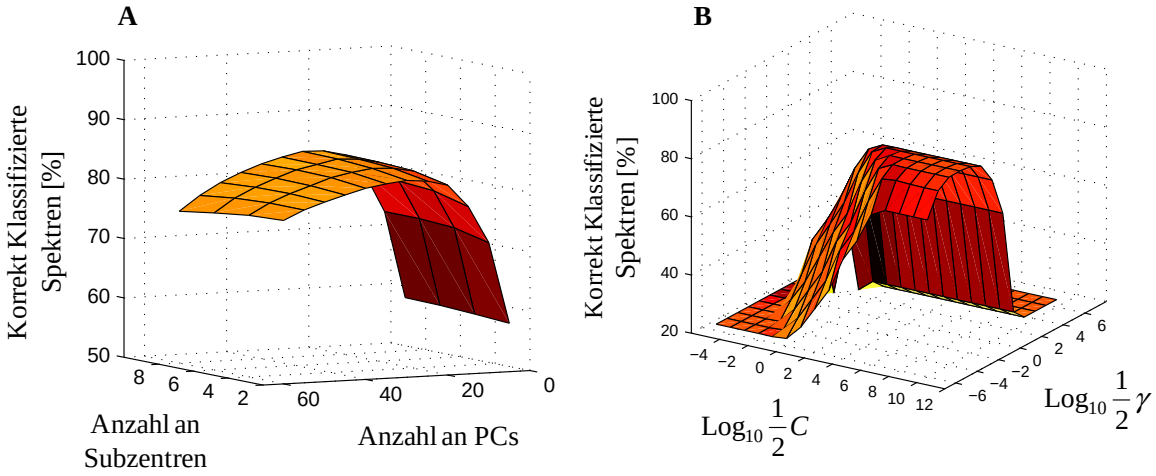


Abbildung 3.14A: „Contour-Plot“ (Abbildung der Klassifikationsraten verschiedener Parameterkombinationen bei einer „Bootstrap“-Validierung mit 100 „Bootstrap“-Stichproben) für paarweise MDA. **Abbildung 3.14B**: „Contour-Plot“ für SVMs („Bootstrap“-Validierung, grobe Suche).

Zusammenfassend kann man festhalten, dass sich die Klassifikation auch bei einem größeren Anteil an Testdaten in der Validierung robust verhält. Im Vergleich zur 50-fachen Kreuzvalidierung sinkt beim „Bootstrapping“ die Klassifikationsrate um 2-3% für beide Klassifikationsmethoden (PK-MDA und SVM), was aufgrund des deutlich kleineren Trainingsdatensatzes eine vergleichsweise leichte Abnahme bedeutet. Daneben wird die Spannweite der gewählten „Tuning“-Parameter bei dem doppelten Validierungsschema größer. Jedoch bleiben die gewählten „Tuning“-Parameter in einem Rahmen, in dem sich die

Klassifikationsergebnisse bei Verwendung einer einfachen Validierung nicht signifikant unterscheiden.

3.4.5 Einfluss der Kultivierungsbedingungen auf die Klassifikation

Die Wachstumsbedingungen von Mikroorganismen beeinflussen deren Zellzusammensetzung und ihren Stoffwechselzustand, was sich in den gemessenen Raman-Spektren durch Variationen innerhalb eines Stammes bemerkbar macht. Da es das Ziel dieser Arbeit ist, einzelne Bakterien direkt ohne vorherige Kultivierung zu identifizieren, muss sichergestellt werden, dass unterschiedliche Wachstumsbedingungen den Klassifikationserfolg nicht beeinträchtigen. Um den Einfluss der Wachstumsparameter auf die Klassifikation einschätzen zu können, wurden in dieser Arbeit die Bakterien unter verschiedensten Bedingungen bzgl. Medium, Temperatur und Zeit kultiviert. Mit Hilfe einer Clusteranalyse sollte nun herausgefunden werden, wie sich die Datenstruktur innerhalb der einzelnen Klassen in Abhängigkeit von den Kultivierungsmedien verhält. Da die MDA auf einer Clusteranalyse mittels „Gaussian Mixtures“ basiert, bietet sie neben der Möglichkeit zur Klassifikation ein geeignetes Interpretationswerkzeug zur Evaluation der Beziehungen zwischen Kultivierungsparametern und Datenstruktur. Zur Untermauerung der Ergebnisse der MDA wurde anschließend die Datenstruktur durch eine SOM veranschaulicht.

3.4.5.1 "Gaussian Mixtures" und die empirische bedingte Entropie

Mit Hilfe der Clustereigenschaften der MDA wurde zunächst der Einfluss der Kultivierungsmedien auf die Verteilung der Daten innerhalb der Klassen untersucht. Dazu wurde ein MDA Modell (30 PCs, 4 Subzentren) auf Basis des gesamten Datensatzes (3642 Spektren) trainiert. Die Bakterienstämme, die auf zwei verschiedenen Wachstumsmedien kultiviert wurden (12 Stämme der Gattung *Staphylococcus* und 7 Stämme der Art *Echericia coli*) wurden für die Analyse herangezogen. Für jede Klasse (hier Bakterienstamm) wurde eine Kontingenztafel erstellt, welche die Anzahl der Spektren wiedergibt, die den einzelnen Subzentren der MDA (SZ1, SZ2, SZ3 und SZ4) zugeteilt sind. Gleichzeitig zeigt sie, wie viele davon zu den jeweiligen Wachstumsmedien gehören (siehe Tabelle 3.10).

Tabelle 3.10: Kontingenztafel und empirische bedingte Entropie (ECE) zur Beschreibung der Übereinstimmeung zwischen Kultivierungsmedien und den 4 Subzentren (SZ1, SZ2, SZ3, SZ4) der MDA.

Name	Kultivierungs-medium	SZ1	SZ2	SZ3	SZ4	ECE
S. cohnii DSM 20260	CA	7	4	9	1	0.7
	CASO	12	3	5	23	
S. cohnii DSM 6669	CA	0	0	21	0	0.0
	CASO	16	3	0	22	
S. cohnii DSM 6718	CA	18	0	3	0	0.0
	CASO	0	15	0	25	
S. cohnii DSM 6719	CA	0	0	21	0	0.1
	CASO	14	4	1	21	
S. epidermidis DSM 1798	CA	0	42	0	10	0.5
	CASO	19	15	21	5	
S. epidermidis 195	CA	0	0	0	39	0.0
	CASO	7	8	20	0	
S. epidermidis DSM 20042	CA	36	20	0	0	0.5
	CASO	21	0	14	15	
S. epidermidis DSM 3269	CA	0	47	0	0	0.3
	CASO	13	6	14	13	
S. epidermidis DSM 3270	CA	0	51	0	0	0.5
	CASO	19	17	14	9	
S. epidermidis ATCC35984	CA	14	127	283	101	0.6
	CASO	114	58	13	95	
S. warneri DSM 20036	CA	4	13	15	11	0.6
	CASO	0	0	22	0	
S. warneri DSM 20316	CA	3	13	0	29	0.2
	CASO	0	3	19	0	
E. coli DSM 1058	NA	1	16	2	11	0.9
	S-1-NA	8	15	7	8	
E. coli DSM 2769	NA	0	20	15	17	0.6
	S-1-NA	28	7	20	1	
E. coli DSM 423	NA	10	19	2	40	0.4
	S-1-NA	0	16	25	0	
E. coli DSM 429	NA	3	28	2	12	0.8
	S-1-NA	7	13	18	7	
E. coli DSM 498	NA	34	0	3	7	0.5
	S-1-NA	3	5	28	6	
E. coli DSM 499	NA	0	4	7	31	0.5
	S-1-NA	1	17	23	0	
E. coli DSM 613	NA	2	30	3	10	0.8
	S-1-NA	6	9	6	28	

Mit Hilfe der Kontingenztafel wurde die empirische bedingte Entropie (ECE) (siehe Kapitel 2.3.5.1) berechnet, die ein quantitatives Maß für die Überlappung von Klassen und Clustern (d.h. Kultivierungsmedium und Subzentrum der MDA) darstellt.

Aus Tabelle 3.10 geht hervor, dass deutliche Übereinstimmungen zwischen der Zugehörigkeit zu den Kultivierungsmedien (Klassen) und der Zuteilung zu den Subzentren der MDA (Clustern) bestehen. Einige Kultivierungsmedien sind durch die vier Subzentren der MDA gut getrennt (ECE<0.5) (z. B. *S. cohnii* DSM 6669, *S. cohnii* DSM 6718, *S. cohnii* DSM 6719, *S. epidermidis* 195, *S. epidermidis* DSM 3269, *S. warneri* DSM 20316). Andere wie die Stämme von *E. coli* zeigen eine hohe Entropie, d.h. die Auftrennung in die Medien durch die Subzentren der MDA ist nicht deutlich erkennbar (ECE>0.5).

Diese Ergebnisse deuten darauf hin, dass für manche Bakterienstämme wie *S. cohnii* DSM 6669, *S. cohnii* DSM 6718, *S. cohnii* DSM 6719, *S. epidermidis* 195, *S. epidermidis* DSM 3269 und *S. warneri* DSM 20316 der Wechsel von einem Kultivierungsmedium zu einem anderen Veränderungen im Metabolismus und der Zellzusammensetzung mit sich bringt, was sich in den Raman-Spektren widerspiegelt. Auf der anderen Seite können nur schwache Übereinstimmungen der Cluster mit den Kultivierungsmedien von *E. coli* gefunden werden, so dass die Zellzusammensetzung bei diesen Stämmen wahrscheinlich nicht so stark von den Wachstumsmedien abhängt.

Die Tatsache, dass sich einige Kultivierungsmedien gut durch die 4 Subzentren der MDA auftrennen lassen, bestätigt die Annahme, dass wechselnde Kultivierungsbedingungen für manche Klassen (Bakterienstämme) streuende Gruppen im Datenraum hervorrufen. Aus der Charakterisierung der Datenstruktur kann man Folgerungen für den Klassifikationserfolg der verschiedenen Klassifikationstechniken ableiten. So wurde in Kapitel 3.4.2 gezeigt, dass nichtlineare Methoden wie SVM und paarweise MDA sehr gute Klassifikationsraten aufweisen und dadurch Klassifikatoren überlegen sind, die nicht in der Lage sind, sich nichtlinearen und streuenden Klassen anzupassen. Dies kann man auf die beschriebene Bildung von Clustern innerhalb der Klassen zurückführen.

Die Betrachtung des Stammes *S. epidermidis* ATCC 35984, für den eine Serie an Messungen durchgeführt wurde, erlaubt eine detailliertere Untersuchung der Beziehungen zwischen Wachstumsbedingungen und Datenstruktur. Tabelle 3.11 zeigt die Kontingenztafel und die entsprechenden ECE-Werte für *S. epidermidis* ATCC 35984 unter Berücksichtigung von

Kultivierungsmedium, Temperatur und Wachstumsdauer. In Tabelle 3.12 ist die Kontingenztafel aus Tabelle 3.11 detaillierter dargestellt.

Tabelle 3.11: Kontingenztafel und empirische bedingte Entropie (ECE) für den Stamm *S. epidermidis* ATCC 35984 unter Berücksichtigung von Kultivierungsmedium, Temperatur und Wachstumsdauer sowie der Verteilung der Spektren auf die 4 Subzentren (SZ1, SZ2, SZ3, SZ4) der MDA.

		SZ1	SZ2	SZ3	SZ4	ECE
Kultivierungsmedium	CA	14	127	283	101	0.6
	CASO	114	40	11	95	
Temperatur	30° C	3	81	113	67	0.8
	37° C	125	86	181	129	
Wachstumsdauer	0-12h	0	129	1	0	0.2
	18-72h	128	38	293	196	

Aus beiden Tabellen wird deutlich, dass die Datenstruktur sehr stark von der Wachstumsdauer beeinflusst ist. So nimmt die ECE bezüglich der Wachstumsdauer einen Wert von kleiner als 0.5 an. (siehe Tabelle 3.11). Dieser Wert ist nachvollziebar, wenn man Tabelle 3.12 genauer betrachtet. Bei einer Wachstumsdauer von 6h bis 12h befinden sich alle Spektren in Subzentrum 2 (SZ2) unabhängig von dem Kultivierungsmedium und der Temperatur (siehe Tabelle 3.12). Zwischen 18h und 72h verteilen sich die Spektren auf die Subzentren SZ1, SZ3 und SZ4. Diese Aufteilung ist nicht zufällig, sondern hängt von dem verwendeten Kultivierungsmedium ab. Das Kultivierungsmedium CA befindet sich hauptsächlich in SZ3, während CA sich in späteren Wachstumsphasen in SZ4 ansiedelt. Das Medium CASO ist auf SZ1 und SZ4 verteilt. Da die Verteilung der Spektren auf die vier Subzentren gleichzeitig von Kultivierungsmedium und Wachstumsdauer abhängt, nimmt der ECE Wert für das Kultivierungsmedium einen mäßig kleinen Wert an (ECE = 0.6)(siehe Tabelle 3.11). Trotzdem ist eine starke Beeinflussung der Datenstruktur durch das Kultivierungsmedium vorhanden. Für die Wachstumstemperatur wird ein geringerer Einfluss auf die Datenstruktur festgestellt (ECE = 0.8) als für die anderen Parameter (Medium und Wachstumsdauer).

Tabelle 3.12: Detaillierte Kontingenztafel für den Stamm *S. epidermidis* ATCC 35984 unter Berücksichtigung von Kultivierungsmedium, Temperatur und Wachstumsdauer sowie der Verteilung der Spektren auf die 4 Subzentren (SZ1, SZ2, SZ3, SZ4) der MDA.

Anzahl der Spektren	Wachstums-dauer	Temperatur	Kultivierungs-medium	SZ1	SZ2	SZ3	SZ4
20	6 h	37°C	CASO	0	20	0	0
21	6 h	37°C	CA	0	21	0	0
27	6 h	30°C	CA	0	27	0	0
20	12 h	37°C	CASO	0	20	0	0
20	12 h	37°C	CA	0	19	1	0
22	12 h	30°C	CA	0	22	0	0
21	18 h	37°C	CASO	14	0	0	7
20	18 h	37°C	CA	0	6	13	1
21	18 h	30°C	CA	0	21	0	0
22	24 h	37°C	CASO	8	0	0	14
22	24 h	37°C	CA	0	0	22	0
22	24 h	30°C	CA	1	11	10	0
23	30 h	37°C	CASO	14	0	0	9
23	30 h	37°C	CA	0	0	23	0
21	30 h	30°C	CA	2	0	16	3
21	36 h	37°C	CASO	4	0	0	17
22	36 h	37°C	CA	0	0	22	0
22	36 h	30°C	CA	0	0	22	0
23	42 h	37°C	CASO	19	0	3	1
22	42 h	37°C	CA	1	0	16	5
21	42 h	30°C	CA	0	0	17	4
20	48 h	37°C	CASO	16	0	1	3
24	48 h	37°C	CA	2	0	21	1
21	48 h	30°C	CA	0	0	13	8
21	54 h	37°C	CASO	8	0	1	12
22	54 h	37°C	CA	5	0	16	1
21	54 h	30°C	CA	0	0	13	8
23	60 h	37°C	CASO	9	0	1	13
21	60 h	37°C	CA	0	0	8	13
23	60 h	30°C	CA	0	0	13	10
23	66 h	37°C	CASO	15	0	5	3
22	66 h	37°C	CA	0	0	17	5
21	66 h	30°C	CA	0	0	7	14
23	72 h	37°C	CASO	7	0	0	16
22	72 h	37°C	CA	3	0	11	8
22	72 h	30°C	CA	0	0	2	20

Alle untersuchten Wachstumsparameter (Medium, Temperatur, Wachstumsdauer) bilden also Gruppen innerhalb der Klassen. Diese Gruppen überlappen zum Teil, was dadurch bedingt sein kann, dass noch mehr Faktoren an der Bildung der Cluster beteiligt sind. Da die natürliche Umgebung von Bakterien von verschiedensten Wachstumseinflüssen geprägt ist, sollte in Betracht gezogen werden, dass die Zuverlässigkeit der Klassifikation reduziert sein kann, wenn nicht alle Wachstumsfaktoren durch den Trainingsdatensatz abgedeckt sind. Für die Anwendung dieser Technik im industriellen Umfeld sollte deshalb darauf geachtet werden, dass im Trainingsdatensatz möglichst viele Wachstumsparameter vertreten sind.

3.4.5.2 "Self Organizing Maps" (SOMs)

SOMs sind sehr gut geeignet, um die Ergebnisse, die mittels MDA und der empirischen bedingten Entropie (ECE) erhalten wurden (Kapitel 3.4.5.1), zu veranschaulichen. In Abbildung 3.15A sind beispielhaft die Spektren aller *S. epidermidis* Stämme auf der über den gesamten Datensatz trainierten SOM dargestellt. Aus Gründen der Übersichtlichkeit, sind die übrigen Bakterienstämme nicht gezeigt.

Es ist deutlich zu sehen, dass die meisten Stämme streuende Gruppen im Raum bilden (siehe Abbildung 3.15). Dies gilt nicht für den Stamm *S. epidermidis* ATCC 35984, für den eine große Anzahl an verschieden kultivierten Bakterien aufgenommen wurde. Stattdessen deckt *S. epidermidis* ATCC 35984 einen großen Bereich auf der SOM ab. Betrachtet man die Wiedererkennungsraten des Stammes *S. epidermidis* ATCC 35984 (siehe Tabelle 3.13), so fällt auf, dass LDA (94.8%) fast genauso gut abschneidet wie die nichtlineare MDA (95.9%). Eine mögliche Erklärung dafür ist, dass keine streuenden Gruppen vorhanden sind (d.h. alle Spektren befinden sich in angrenzenden Feldern auf der SOM). Deswegen bringt die Eigenschaft der MDA, streuende Gruppen modellieren zu können, hier keine Vorteile.

Durch die paarweise Klassifikation (PK-LDA, PK-MDA) steigen die Klassifikationsraten auf 99.1% an, da dadurch die Nichtlinearität der Klassenstruktur besser abgebildet werden kann. Die anderen Stämme, für die weniger Spektren aufgenommen wurden und die zudem weniger divers (bzgl. Wachstumsdauer) sind, zeigen streuende Cluster auf der SOM. Dies führt dazu, dass die MDA in den meisten Fällen bessere Klassifikationsraten erzielt als die LDA. Da die Klassen jedoch stark überlappen (besonders die Stämme DSM 20042, DSM

3269, and DSM 3270), sind sie sowohl durch eine LDA als auch durch eine MDA schwer zu klassifizieren, so dass MDA für manche Stämme nicht besser abschneidet als LDA.

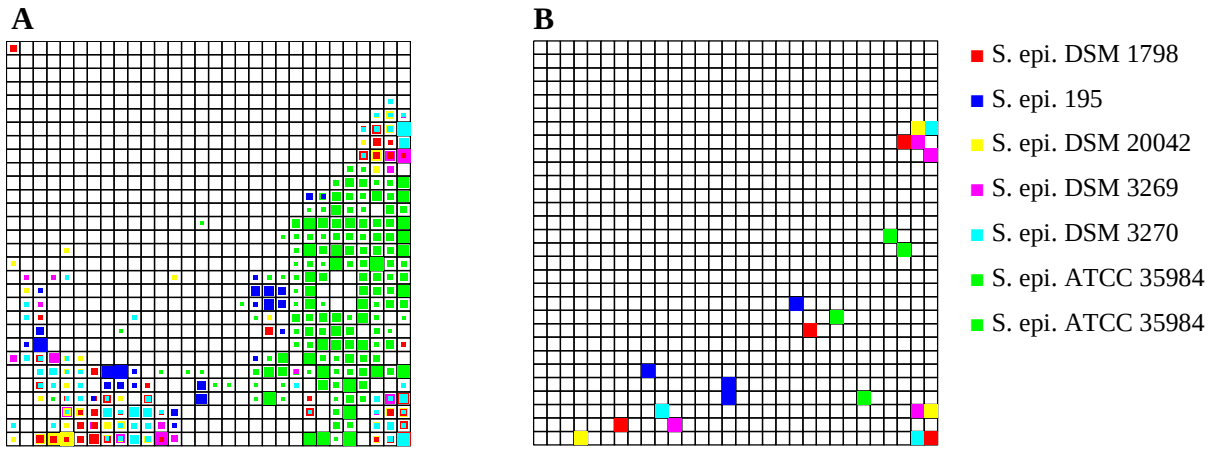


Abbildung 3.15A: SOM mit verschiedenen Stämmen von *S. epidermidis* (DSM 1798, 195, DSM 20042, DSM 3269, DSM 3270, ATCC 35984) (Verwendung von 30 PCs). **Abbildung 3.15B**: SOM mit den Zentroiden der vier Subzentren von MDA für die Stämme von *S. epidermidis*

Tabelle 3.13: Klassifikationsraten (%) von LDA (30 PCs), paarweise LDA (PK-LDA) (30 PCs), MDA (30 PCs) und paarweise MDA (PK-MDA) (30 PCs) für verschiedene Stämme von *S. epidermidis*.

	LDA	**PK-LDA**	**MDA**	**PK-MDA**
S. epidermidis DSM 1798	27.7%	62.5%	43.8% ↑	64.3% ↑
S.epidermidis 195	75.7%	82.4%	89.2% ↑	90.5% ↑
S.epidermidis DSM 20042	45.3%	60.4%	45.3% ↔	63.2% ↑
S.epidermidis DSM 3269	53.8%	46.2%	47.3% ↓	50.5% ↑
S.epidermidis DSM 2270	43.6%	60.9%	58.2% ↑	66.4% ↑
S.epidermidis ATCC 35984	94.8%	99.1%	95.9% ↑	99.1% ↔

In Kapitel 3.4.5.1 wurde gezeigt, dass sich die verwendeten Kultivierungsparameter gut in den 4 Subzentren der MDA wiederfinden lassen. Auch diese Beobachtung kann durch eine SOM visualisiert werden (siehe Abbildung 3.16). Aus Gründen der Übersichtlichkeit ist auf dieser SOM der Stamm *S. epidermidis* ATCC 35984 nicht dargestellt.

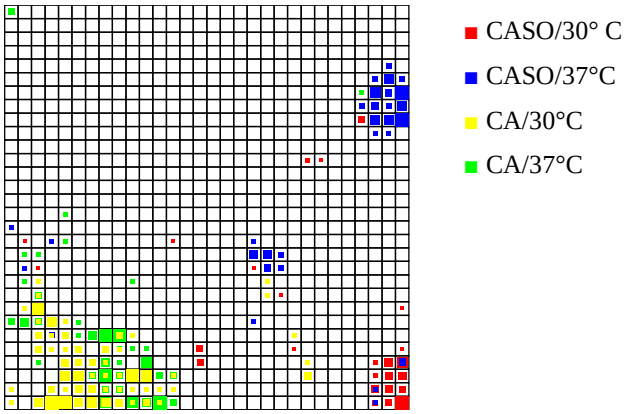


Abbildung 3.16: SOM mit den Raman-Spektren von verschiedenen *S. epidermidis* Stämmen (DSM 1798, 195, DSM 20042, DSM 3269, DSM 3270), die unter verschiedenen Bedingungen kultiviert wurden (CASO/30°C, CASO/37°C, CA/30°C, CA/37°C)

Abbildung 3.16 zeigt erneut, dass sich Gruppen in Abhängigkeit von den Kultivierungsparametern bilden. Die Spektren, die von Bakterien stammen die auf CASO kultiviert wurden, sind klar von den Spektren getrennt, die auf eine Kultivierung mit CA zurückgehen. Ebenso bildet die Temperatur bei der Kultivierung auf dem Kultivierungsmedium CASO deutliche Cluster. Für das Kultivierungsmedium CA wird dagegen kein Effekt bezüglich der Temperatur gefunden.

Die Clusteranalyse mittels MDA sowie mit Hilfe von Kohonenkarten konnte in dieser Arbeit dazu genutzt werden, um den Effekt der verschiedenen Kultivierungsparameter auf die Datenstruktur und den Klassifikationserfolg bei der Differenzierung von einzelnen Bakterienzellen zu analysieren. Dabei stellte sich heraus, dass alle Wachstumsparameter Cluster im Datenraum bilden. Die Ausprägung dieser Cluster für einzelne Stämme konnte

mit der Klassifikationsrate verschiedener linearer und nichtlinearer Klassifikationsmethoden in Zusammenhang gebracht werden. Als Fazit der Analyse kann man festhalten, dass die durch unterschiedliche Wachstumsparameter bedingte Clusterbildung innerhalb der Klassen dazu führt, dass Klassifikationsmethoden, die nur lineare Entscheidungsgrenzen bilden können, bei der Klassifikation schlechter abschneiden, als Methoden, die sich flexiblen und streuenden Klassen anpassen können. Außerdem kann man darauf schließen, dass die Zuverlässigkeit der erhaltenen Klassifikationsraten für zukünftige Vorhersagen nur dann gegeben ist, wenn alle möglichen Wachstumszustände eines Bakteriumstammes im Trainingsdatensatz vertreten sind. In dem hier untersuchten Datensatz stellt der Stamm *S. epidermidis* ATCC 35984 ein Beispiel für das Abdecken der möglichst gesamten Heterogenität eines Bakterienstammes dar. Für diesen Stamm wurde eine große Menge an Bakterien, die unter verschiedenen Parametern und dabei vor allem unter verschiedensten Wachstumszeiten kultiviert wurden, in den Datensatz aufgenommen. Es wurde gezeigt, dass die Intra-Klassen-Gruppen dieses Stammes nicht deutlich getrennt voneinander vorliegen wie bei vielen anderen Klassen. Deshalb zeigt die Klassifikation dieses Stammes sowohl mit linearen (94.8%) als auch mit nichtlinearen Klassifikationsmethoden (99.1%) eine sehr hohe Wiedererkennungsrate (siehe Tabelle 3.13). Andere Bakterienstämme zeigen möglicherweise ein anderes Verhalten. Jedoch demonstriert die Studie, dass die Etablierung eines Trainingsdatensatzes der möglichst die gesamte natürliche Heterogenität von Mikroorganismen abdeckt, grundlegend für eine zuverlässige Identifizierung von einzelnen Bakterienzellen ohne vorherige Kultivierung ist. Möglicherweise können die hier erhaltenen vielversprechenden Klassifikationsergebnisse sogar noch verbessert werden, indem die Diversität des Datensatzes dahingehend erhöht wird, dass alle Bakterienstämme in einer ähnlich hohen Diversität bezüglich der Wachstumsparameter vorliegen wie der Stamm *S. epidermidis* ATCC 35984.

3.4.6 Vorhersage von unbekannten Testdaten und Ausreißer-Erkennung

Für das „Online-Monitoring“ in industriellen Reinräumen reicht es nicht aus, die Bakterien durch Klassifikation zu einem der Bakterienstämme des Trainingsdatensatzes zuzuordnen. Auch neue Spektren, die von Bakterien stammen, die nicht im Trainingsdatensatz enthalten sind (Vorhersageausreißer) müssen als solche erkannt werden. Diese sind unter Verwendung des Trainingsdatensatzes nicht klassifizierbar und müssen mit Hilfe weiterführender Untersuchungen auf ihre Identität überprüft werden. Die Klassifikationsmethoden MDA und SVMs, die in den vorhergehenden Kapiteln sehr gute Erfolge bei der Klassifikation der Spektren gezeigt haben, werden im Folgenden dazu verwendet Vorhersageausreißer zu erkennen.

3.4.6.1 Ausreißererkennung auf Basis der MDA

Bei der MDA wird üblicherweise die unbedingte Wahrscheinlichkeitsdichte $p(\mathbf{x})$ als Maß für die Unbekanntheit eines Spektrums und somit für die Erkennung von Vorhersageausreißern herangezogen. Daneben ist es auch möglich, die klassenbedingte Wahrscheinlichkeitsdichte $p(\mathbf{x}|C_j)$ zu verwenden. Für beide Ansätze wurde zunächst eine Kreuzvalidierung durchgeführt, bei der jeder Stamm einmal als Testset aus dem Datensatz entnommen wurde. Auf Basis der Trainingsdaten wurden die Grenzwerte (s bzw. s_j) für $p(\mathbf{x})$ und $p(\mathbf{x}|C_j)$ festgelegt (siehe Kapitel 2.3.3.4.4). Wenn die berechneten Werte $p(\mathbf{x})$ bzw. $p(\mathbf{x}|C_j)$ der Testdaten, diese Grenzwerte unterschritten, wurden sie als Ausreißer detektiert. Idealerweise sollten bei der Kreuzvalidierung alle Testdatenspektren als Ausreißer erkannt werden, da bei jedem Kreuzvalidierungsschritt der gesamte Stamm aus dem Datensatz entfernt wurde. Um zudem die Leistungsfähigkeit der Methoden auf Artebene zu erfassen, wurden bei der Kreuzvalidierung neben dem Testset, das aus einem Bakterienstamm besteht, zusätzlich alle Stämme derselben Art aus dem Trainingsdatensatz entfernt.

Der Anteil der erkannten Vorhersageausreißer ist abhängig von der definierten Irrtumswahrscheinlichkeit, die durch die Variable z beschrieben wird. Je größer z ist, desto mehr Ausreißer werden erkannt. Gleichzeitig nimmt aber auch der Anteil an fälschlicherweise detektierten Ausreißern (falsch Positive) zu. Die erwartete falsch-positiv-Rate ist also z%. In Abbildung 3.17 ist der Prozentsatz der detektierten Vorhersageausreißer

in der Kreuzvalidierung gegen die Irrtumswahrscheinlichkeit aufgetragen. Daraus wird ersichtlich, dass die klassenbedingten Wahrscheinlichkeitsdichten $p(\mathbf{x}|C_j)$ gegenüber den bedingten Wahrscheinlichkeitsdichten $p(\mathbf{x})$ bei der Ausreißererkennung von Vorteil sind. Dies ist leicht nachvollziehbar, da $p(\mathbf{x}|C_j)$ im Gegensatz zu $p(\mathbf{x})$ die Struktur der einzelnen Klassen in die Berechnung einbezieht. Da der Bakteriendatensatz sehr heterogen ist, wirkt sich die Berücksichtigung der klassenspezifischen Merkmale unter Verwendung von $p(\mathbf{x}|C_j)$ positiv auf die Ausreißerdetektion aus.

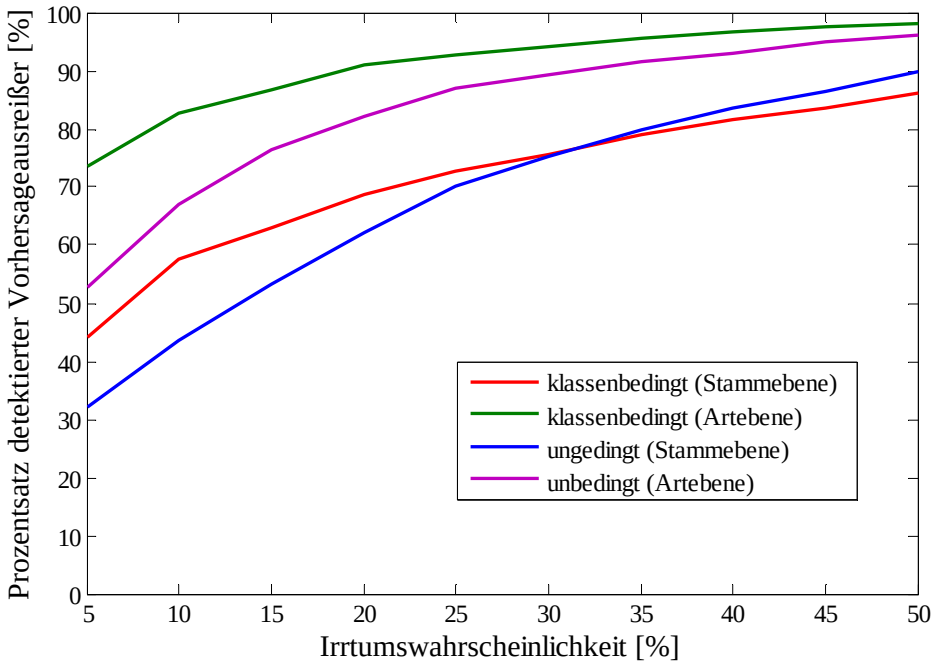


Abbildung 3.17: Detektion von Vorhersageausreißern mittels $p(\mathbf{x})$ (unbedingte Wahrscheinlichkeitsdichte) und $p(\mathbf{x}|C_j)$ (klassenbedingte Wahrscheinlichkeitsdichte) auf Stamm- und auf Artebene bei verschiedenen Werten der Irrtumswahrscheinlichkeit.

Die Wahl der Irrtumswahrscheinlichkeit hängt davon ab, wie gravierend die Folgen einer Fehlklassifikation sind. Sind die Folgen schwer, sollte eine höhere Irrtumswahrscheinlichkeit gewählt werden, so dass möglichst alle Vorhersageausreißer als solche erkannt werden. Dabei muss in Kauf genommen werden, dass auch ein höherer Anteil an Spektren, die keine

Ausreißer sind, als solche deklariert werden (falsch Positive). Weiterführende Untersuchungen müssen dann Aufschluss über die Identität der gefundenen Ausreißer geben. In dieser Studie wurde angenommen, dass vor allem das Nichterkennen von Ausreißern auf Artebene gravierende Folgen mit sich bringt. Deshalb wurde eine Irrtumswahrscheinlichkeit von 20% gewählt. Auf diese Weise wurden durch die klassenbedingten Wahrscheinlichkeitsdichten $p(\mathbf{x}|C_j)$ in der Kreuzvalidierung 91% der Testdaten auf Artebene als Ausreißer detektiert, was hier als ausreichend definiert wurde. Die unbedingte Wahrscheinlichkeitsdichte lieferte bei dieser Irrtumswahrscheinlichkeit einen Wert von 82.2%.

Für beide Ansätze (unbedingte und klassenbedingte Wahrscheinlichkeitsdichte) sind in den folgenden Tabellen die Prozentsätze der detektierten Ausreißer bei der Kreuzvalidierung für jeden einzelnen Stamm aufgelistet. Dabei erkennt man, dass für beide Methoden die Detektionsraten auf Stammebene sehr stark variieren; d.h. manche Stämme werden schlecht als Ausreißer detektiert, andere wiederum sehr gut. Auf Artebene stellt man fest, dass die Varianz bei Verwendung der klassenbedingten Wahrscheinlichkeitsdichten wesentlich geringer ist als bei der Verwendung der unbedingten Wahrscheinlichkeitsdichten. Fast alle Stämme werden auf Artebene mit Hilfe der klassenbedingten Wahrscheinlichkeitsdichte sehr gut erkannt. Nur 2 Stämme zeigen eine etwas schwächere Detektionsrate (*B. subtilis* DSM 347 und *S. epidermidis* DSM 44195).

Tabelle 3.14: Detektion von Vorhersageausreißern unter Verwendung der unbedingten Wahrscheinlichkeitsdichten der MDA (4 Subzentren, 30 PCs). Durch Kreuzvalidierung wurde jeder Bakterienstamm genau einmal aus dem Datensatz entnommen und als Testset verwendet. Die 1. Spalte der Tabelle zeigt den Prozentsatz an Spektren eines Stammes, der bei der Kreuzvalidierung als Ausreißer detektiert wurde (Irrtumswahrscheinlichkeit: 20%). In Spalte 2 der Tabelle sind die Ergebnisse auf Artebene gezeigt. Dabei wurden bei der Kreuzvalidierung neben dem jeweiligen Stamm, der als Testset diente, alle Stämme der gleichen Art pro Validierungsschritt aus dem Trainingsdatensatz herausgelassen.

Name	Korrekt erkannte Vorhersageausreißer auf Stammebene (%)	Korrekt erkannte Vorhersageausreißer auf Artebene (%)
B. pumilus DSM 27	89.5	94.7
B. pumilus DSM 361	85.5	92.8
B. sphaericus DSM 28	88.7	88.7
B. sphaericus DSM 396	69.0	88.1
B. subtilis DSM 10	93.3	95.4
B. subtilis DSM 347	64.3	73.8
M. luteus DSM 20030	87.5	93.8
M. luteus DSM 348	99.4	99.7
M. lylae DSM 20315	100.0	100.0
M. lylae DSM 20318	60.0	50.0
S. cohnii DSM 20260	32.8	48.4
S. cohnii DSM 6669	33.9	41.9
S. cohnii DSM 6718	19.7	39.3
S. cohnii DSM 6719	13.1	26.2
S. epidermidis DSM 1798	59.8	92.9
S. epidermidis DSM 44195	20.3	47.3
S. epidermidis DSM 20042	66.0	96.2
S. epidermidis DSM 3269	49.5	94.6
S. epidermidis DSM 3270	50.9	95.5
S. epidermidis ATCC 35984	44.2	64.2
S. warneri DSM 20036	12.3	32.3
S. warneri DSM 20316	31.3	46.3
E. coli DSM 1058	45.6	98.5
E. coli DSM 2769	55.6	97.2
E. coli DSM 423	74.1	100.0

E. coli DSM 429	40.0	97.8
E. coli DSM 498	60.5	100.0
E. coli DSM 499	50.6	100.0
E. coli DSM 613	34.0	98.9
Mittelwert (%)	56.3	79.1
Prozentsatz an richtig detektierten Vorhersageausreißern	62.3	82.2

Tabelle 3.15: Detektion von Vorhersageausreißern unter Verwendung der klassenbedingten Wahrscheinlichkeitsdichten der MDA (4 Subzentren, 30 PCs). Durch Kreuzvalidierung wurde jeder Bakterienstamm genau einmal aus dem Datensatz entnommen und als Testset verwendet. Die 1. Spalte der Tabelle zeigt den Prozentsatz an Spektren eines Stammes, der bei der Kreuzvalidierung als Ausreißer detektiert wurde (Irrtumswahrscheinlichkeit: 20%). In Spalte 2 der Tabelle sind die Ergebnisse auf Artebene gezeigt. Dabei wurden bei der Kreuzvalidierung neben dem jeweiligen Stamm, der als Testset diente, alle Stämme der gleichen Art pro Validierungsschritt aus dem Trainingsdatensatz herausgelassen.

Name	**Korrekt erkannte Vorhersageausreißer auf Stammebene (%)**	**Korrekt erkannte Vorhersageausreißer auf Artebene (%)**
B. pumilus DSM 27	31.6	73.7
B. pumilus DSM 361	68.1	78.3
B. sphaericus DSM 28	83.0	81.1
B. sphaericus DSM 396	52.4	61.9
B. subtilis DSM 10	90.5	85.9
B. subtilis DSM 347	50.0	47.6
M. luteus DSM 20030	97.9	100.0
M. luteus DSM 348	98.9	100.0
M. lylae DSM 20315	77.8	77.8
M. lylae DSM 20318	95.0	95.0
S. cohnii DSM 20260	84.4	92.2
S. cohnii DSM 6669	79.0	77.4
S. cohnii DSM 6718	54.1	75.4
S. cohnii DSM 6719	60.7	77.0
S. epidermidis DSM 1798	42.9	94.6

S. epidermidis DSM 44195	45.9	56.8
S. epidermidis DSM 20042	37.7	95.3
S. epidermidis DSM 3269	22.6	94.6
S. epidermidis DSM 3270	31.8	96.4
S. epidermidis ATCC 35984	91.2	95.5
S. warneri DSM 20036	40.0	67.7
S. warneri DSM 20316	55.2	79.1
E. coli DSM 1058	23.5	100.0
E. coli DSM 2769	31.5	93.5
E. coli DSM 423	49.1	95.5
E. coli DSM 429	15.6	96.7
E. coli DSM 498	37.2	100.0
E. coli DSM 499	32.5	97.6
E. coli DSM 613	20.2	93.6
Mittelwert (%)	**55.2**	**85.5**
Prozentsatz an richtig detektierten Vorhersageausreißern	**68.8**	**91.0**

Im nächsten Schritt wurde das „verblindete“ Testset, das in Kapitel 3.2 vorgestellt wurde, analysiert. Dieses enthält sowohl „bekannte“ als auch „unbekannte“ Bakterienstämme (d.h. Bakterienstämme, die im Trainingsdatensatz enthalten sind und Bakterienstämme, die nicht im Trainingsdatensatz enthalten sind). Von den „unbekannten“ Bakterienstämmen unterscheiden sich einige von den Bakterienstämmen des Trainingsdatensatzes auf Stammebe (*M. luteus* BCD 3906, *E. coli* DSM 5208 und *E. coli* DSM 426), einige auf Artebene (*S. hominis* BCD 2684, *S. thermophilus* DSM 20617 und *L. acidophilus* DSM 9126). Vor der Klassifikation der „bekannten“ Bakterienstämme wurde eine Detektion der „unbekannten“ Bakterienstämme (Vorhersageausreißer) durchgeführt, wobei als Maß für die Ausreißerdetektion die unbedingte Wahrscheinlichkeitsdichte mit einer Irrtumswahrscheinlichkeit von 20% gewählt wurde. Die Ergebnisse der Ausreißerdetektion sind in Tabelle 3.16 gezeigt.

Tabelle 3.16: Detektion von Bakterienstämmen, die nicht im Trainingsdatensatz enthalten sind (Vorhersageausreißer). Dies erfolgte mit Hilfe der klassenbedingten Wahrscheinlichkeitsdichte $p(\mathbf{x}|C_j)$, die mittels MDA (4 Subzentren, 30 PCs) geschätzt wurde. Vorhersageausreißer wurden mit einer Irrtumswahrscheinlichkeit von 20% detektiert.

Irrtumswahrscheinlichkeit: 20%					
Name			**Detektierte Ausreißer**	**Falsch Negative**	**Falsch Positive**
Micrococcus	*luteus*	BCD 3906	44	1	
Escherichia	*coli*	DSM 5208	22	4	
Escherichia	*coli*	DSM 426	11	13	
Staphylococcus	*hominis*	BCD 2684	18	3	
Streptococcus	*thermophilus*	DSM 20617	25	3	
Lactobacillus	*acidophilus*	DSM 9126	22	3	
Gesamt			**142**	**27**	**28**

Fast alle „unbekannten“ Spektren des „verblindeten“ Testsets (142 von 169) wurden mittels MDA als Vorhersageausreißer detektiert. 27 von 169 „unbekannten“ Bakterienstämmen wurden nicht als Vorhersageausreißer erkannt (falsch Negative). 28 von 130 „bekannten“ Spektren wurden dagegen fälschlicherweise als Ausreißer deklariert (falsch Positive).

Im Anschluss an die Ausreißerdetektion wurden die verbleibenden „bekannten“ Spektren mittels paarweiser MDA klassifiziert. Dabei erhielt man nur wenige Fehlklassifikationen (siehe Tabelle 3.17). 95 von 102 Spektren, die nach der Detektion der Vorhersageausreißer als „bekannt“ zurückblieben, wurden korrekt klassifiziert. 23 von den 28 Spektren, die fälschlicherweise als Ausreißer detektiert wurden (falsch Positive), wären zudem mittels PK-MDA korrekt klassifiziert worden.

Tabelle 3.17: Ergebnisse der Klassifikation des "verblindeten" Testdatensatzes mittels PK-MDA

	Anzahl der Spektren	Anzahl korrekt klassifizierter Spektren
Nicht als Ausreißer detektierte Bakterienstämme	102	95
Fälschlicherweise als Ausreißer detektiere Bakterienstämme	28	23

Die Analyse des „verblindeten" Testsets mittels MDA zeigt erneut, dass MDA sowohl für die Detektion von Vorhersageausreißern als auch für die Klassifikation der Spektren sehr gut geeignet ist. Die Zuverlässigkeit der Ausreißerdetektion ist vor allem auf Artebene sehr hoch. Aus diesem Grumd stellt MDA ein nützliches Werkzeug für die Raman-spektroskopische Identifizierung von Bakterien im industriellen „Online-Monitoring" dar.

3.4.6.2 Ausreißererkennung auf Basis von SVMs

Die Erkennung von Vorhersageausreißern ist mit Hilfe einer Ein-Klassen-SVM (engl. „One-Class SVM") möglich. Analog zur MDA wurde die Ein-Klassen-SVM verwendet, um die Vorhersageausreißer des „verblindeten" Testsets zu detektieren. Dabei wurden zunächst anhand des Datensatzes die Parameter für die Ausreißerdetektion (ν und γ des RBF-Kernels, siehe Kapitel 2.3.3.4.6) festgelegt. Dazu wurden Kreuzvalidierungen mit verschiedenen Werten von ν und γ durchgeführt, bei denen schrittweise jeweils ein Stamm bzw. eine Art aus dem Datensatz entnommen wurden. Zudem wurde ein kleiner Anteil der Spektren (10% pro Bakterienstamm) aus dem Trainingsdatensatz entfernt, um die Rate der fälschlicherweise detektierten Ausreißer (falsch Positive) zu bestimmen. Der Prozentsatz der korrekt detektierten Ausreißer sowie die falsch-positiv-Rate wurden registriert. Es stellte sich heraus, dass in dem Bereich von $\gamma \in (2 \cdot 10^{-7}; 2 \cdot 10^{-3})$ falsch-positiv-Raten erhalten werden, die sehr gut mit den Werten von ν übereinstimmen. Bei verschiedenen γ-Werten waren innerhalb dieses Bereiches keine wesentlichen Unterschiede bezüglich des Prozentsatzes der

detektierten Ausreißer zu erkennen. Für die folgenden Berechnungen wurde γ gleich $2 \cdot 10^{-6}$ gesetzt. In Abbildung 3.18 ist jeweils der Prozentsatz der detektierten Vorhersageausreißer auf Stamm- und Artebene gegen verschiedene Werte von ν bzw. der Irrtumswahrscheinlichkeit aufgetragen.

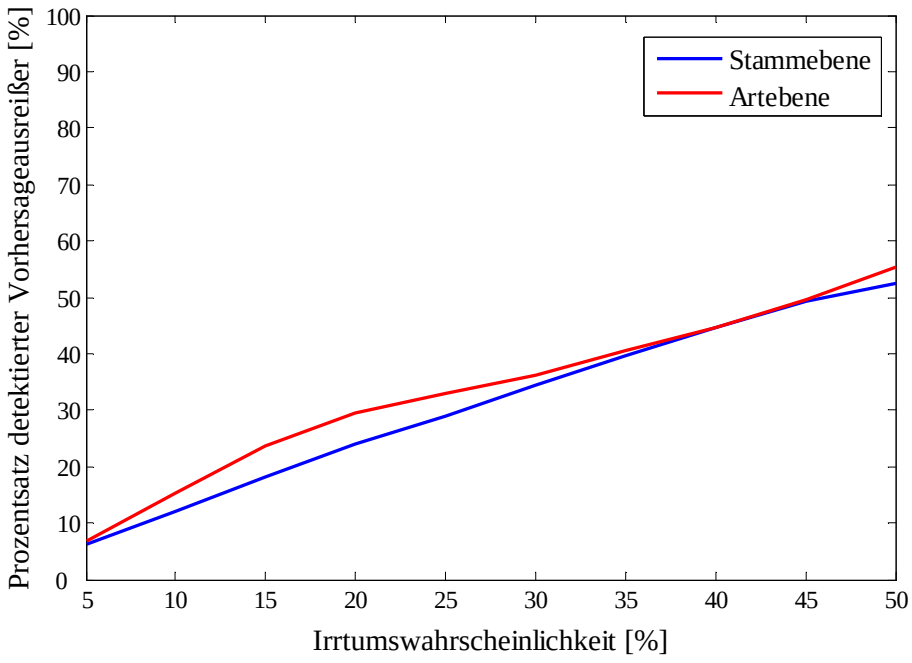


Abbildung 3.18: Detektion von Vorhersageausreißern auf Stamm- und Artebene mittels Ein-Klassen-SVM (RBF-Kernel; $\gamma = 2 \cdot 10^{-6}$) bei verschiedenen Werten von ν (Irrtumswahrscheinlichkeit).

Aus der Abbildung wird deutlich, dass der Prozentsatz der detektierten Vorhersageausreißer annähernd der falsch-positiv-Rate entspricht, was die Methode für die Ausreißerdetektion unbrauchbar macht. Auf Artebene erhält man zwar leicht bessere Resultate als auf Stammebene. Allerdings reicht auch dieses für eine Ausreißerdetektion nicht aus. Die Ein-Klassen-SVM mit einem RBF-Kernel ohne vorherige Variablenselektion ist also für die Detektion von Vorhersageausreißern für diese Aufgabenstellung nicht geeignet. Da die Leistung einer Ein-Klassen-SVM von dem verwendeten Kernel abhängt, können die

Ergebnisse möglicherweise durch einen Wechsel des Kernels verbessert werden. In dieser Arbeit wurde eine möglichst einfach durchführbare Methode zur Ausreißerdetektion gesucht, die direkt in Verbindung mit der Klassifikation verwendet werden kann. Deshalb wurde der Einsatz verschiedener Kernels hier nicht getestet.
Im Anschluss an die Ausreißerdetektion, wurden die „verblindeten“ Testdaten mit Hilfe von SVMs klassifiziert (siehe Tabelle 3.18). Dabei zeigte sich erneut eine hervorragende Klassifikationsrate. Nur 3 von 130 Spektren wurden unter Verwendung der SVMs falsch klassifiziert.

Tabelle 3.18: Ergebnisse der Klassifikation des “verblindeten” Testdatensatzes mittels SVM

	Anzahl der Spektren	**Anzahl korrekt klassifizierter Spektren**
„bekannte“ Bakterienstämme	130	127

Zusammenfassend kann man festhalten, dass SVMs auch bei der Klassifikation des „verblindeten“ Testsets sehr gut abschneiden. Die Detektion von Vorhersageausreißern ist stark von dem verwendeten Kernel sowie den eingesetzten Parametern abhängig. Mit dem RBF-Kernel, der bei der Klassifikation verwendet wurde, war eine erfolgreiche Ausreißererkennung nicht möglich.

4 Zusammenfassung und Ausblick

Schwingungsspektroskopische Verfahren stellen eine Alternative zu konventionellen mikrobiologischen Techniken bei der Identifizierung von Mikroorganismen dar. Speziell die konfokale Mikro-Raman-Spektroskopie ist eine Technik, die nur wenig Biomasse benötigt, was die Untersuchung von einzelnen Bakterienzellen ohne vorherige Kultivierung ermöglicht. Sie ist deshalb für Anwendungsgebiete prädestiniert, in denen eine schnelle Identifizierung von Mikroorganismen erforderlich ist. Ein derartiges Einsatzgebiet stellt das „Online-Monitoring“ in der industriellen Reinraumherstellung dar. In diesem Umfeld ist die Anzahl der mikrobiellen Partikel begrenzt und es handelt sich nur um bestimmte Bakterienarten, die in der Analyse berücksichtigt werden müssen. Dennoch muss auch hier eine große Anzahl an Bakterienzellen untersucht werden. Um die konfokale Mikro-Raman-Spektroskopie für diesen Zweck praktisch anwendbar zu machen, ist ein vollautomatischer Ablauf der apparativen Messungen vor Ort erforderlich. Daneben benötigt man ein robustes und zuverlässiges Verfahren zur Auswertung der Spektren, das in das automatisierte System integriert ist. Mit dem zuletzt genannten Punkt beschäftigt sich die vorliegende Dissertation. In dieser Arbeit wurde ein umfassendes datenanalytisches Auswertungssystem entwickelt, das zur Identifizierung von Bakterien mittels Mikro-Raman-Spektroskopie im „Online-Monitoring“ von industriellen Reinräumen geeignet ist. Die Entwicklung des Verfahrens basiert auf der Analyse eines hochdiversen Datensatzes, der 29 Bakterienstämme enthält, die in der Regel in industriellen Reinräumen vorzufinden sind. Neben der Evaluation verschiedener spektraler Vorbehandlungsmethoden wurden insbesondere Klassifikationsmethoden auf ihre Tauglichkeit für die gegebene Aufgabenstellung untersucht. Dabei stellte sich heraus, dass Klassifikationsalgorithmen, die nichtlineare Entscheidungsgrenzen bilden können (Quadratische Diskriminanzanalyse: QDA, „Gaussian Mixture“ Diskriminanzanalyse: MDA, *k*-nächste Nachbarn Klassifizierer: *k*NN und „Support Vector Machines“: SVMs), eine wesentlich bessere Differenzierung der Bakterien ermöglichen als lineare Modelle („Partial least squares“-Diskriminanzanalyse: PLS-DA und Lineare Diskriminanzanalyse: LDA). Um die Signifikanz der Unterschiede zwischen den

Klassifikationsraten zu ermitteln, wurden eine kreuzvalidierte Varianzanalyse (CVANOVA) sowie der Kruskal-Wallis-Test herangezogen.

Zur Verbesserung der Klassifikationsrate wurden die Algorithmen zusätzlich einer paarweisen Klassifikation unterzogen. Dies gilt nicht für SVMs, da bei diesen sowieso eine paarweise Klassifikation stattfindet. Der paarweise Ansatz führt dazu, dass die Flexibilität der Entscheidungsgrenzen für manche Klassifikationsmethoden erhöht wird, was besonders bei nichtlinearen Klassifikationsproblemen gewinnbringend eingesetzt werden kann. Auf diese Weise konnte für PLS-DA, LDA und MDA in dieser Arbeit eine enorme Steigerung an Vorhersagegenauigkeit erreicht werden.

Letztendlich wurden die besten Wiedererkennungsraten mit den Methoden SVM und paarweise MDA erzielt. So erhielt man bei der Analyse des hochdiversen Bakterien-Datensatzes (unterschiedliche Kultivierungsbedingungen der Bakterien) mit Hilfe der paarweisen MDA eine Wiedererkennungsrate von 86.6% (50-fache Kreuzvalidierung), während unter der Verwendung von SVMs 87.3% der Spektren korrekt klassifiziert wurden. Im weiteren Verlauf der Arbeit wurden die beiden Klassifikationsmethoden (paarweise MDA und SVMs) bezüglich ihrer Robustheit, Zuverlässigkeit und Nützlichkeit für die vorliege Aufgabenstellung charakterisiert.

Für beide Techniken erwies sich der Modellselektionsschritt als unkritischer Faktor. Dies wurde überprüft, indem ein doppeltes Validierungsschema bestehend aus zwei Schleifen mit wiederholter Stichprobenziehung durchgeführt wurde. Durch die Generierung von multiplen externen Testsets (äußere Schleife), die unabhängig von der Modellselektion (innere Schleife) sind, konnte demonstriert werden, dass die Verzerrung des Vorhersagefehlers durch die Modellselektion (engl. Model Selection Bias) sehr gering ist; d.h. ein durch Modellselektion verursachtes „Overfitting“ kann hier weitestgehend ausgeschlossen werden. Durch das Verwenden von „Bootstrapping“ in der äußeren Schleife wurde zudem deutlich, dass sich die Klassifikationsmodelle gegenüber der Verwendung kleinerer und unterschiedlich zusammengesetzter Trainingsdaten robust verhalten.

Paarweise MDA bietet aufgrund der geringeren Modellkomplexität eine attraktive Alternative zur Klassifikation mittels SVMs. So werden beispielsweise leicht interpretierbare Modelle erhalten, aus denen Informationen über die vorliegende Datenstruktur extrahiert werden können. Dies wurde genutzt, um die Auswirkungen der Kultivierungsparameter auf

die Verteilung der Daten im Raum innerhalb der einzelnen Klassen (Bakterienstämme) und somit auf die Klassifikation zu analysieren. Diese Information ist für eine zuverlässige Differenzierung von einzelnen Bakterien ohne vorherige Kultivierung entscheidend, da man Bakterien in ihrer natürlichen Umgebung in verschiedenen Wachstumsphasen und Stoffwechselzuständen vorfindet. Es konnte gezeigt werden, dass sich Kultivierungsmedium, Temperatur und Zeit in den Subzentren der MDA wiederfinden lassen. Unterschiedliche Wachstumsparameter führen also offensichtlich zur Bildung von Gruppen im Datenraum. Mit Hilfe von Kohonen-Karten (engl. Self Organizing Maps: SOMs) konnten diese Gruppen, die sich innerhalb der einzelnen Klassen bilden, visualisiert werden. Dass die Gruppen durch die verwendeten Kultivierungsparameter entstehen, konnte ebenfalls auf der SOM sichtbar gemacht werden. Die beschriebene Gruppenbildung innerhalb der Klassen führt zu der Annahme, dass für eine erfolgreiche Klassifikation von einzelnen Bakterienzellen mittels Mikro-Raman-Spektroskopie Klassifikationsmethoden erforderlich sind, die nichtlineare und streuende Klassen modellieren können. Dies steht im Einklang mit der beobachteten Überlegenheit von Klassifikationsalgorithmen, die nichtlineare Entscheidungsgrenzen bilden können. Die Tatsache, dass verschiedene Wachstumsparameter die Bildung von Gruppen im Datenraum innerhalb der Klassen verursachen, lässt außerdem darauf schließen, dass die vielversprechende Wiedererkennungsrate von ca. 87% auf Stammebene für künftige Vorhersagen nur dann reproduzierbar ist, wenn der Trainingsdatensatz möglichst alle Variationen der zu identifizierenden Bakterienstämme abdeckt. Der im Datensatz enthaltene Stamm *S. epidermidis* ATCC 35984, für den eine große Menge Bakterien unter verschiedensten Wachstumsbedingungen kultiviert wurde, stellt ein Beispiel für das Abdecken der maximal möglichen Heterogenität eines Bakterienstammes dar. Bei diesem Stamm verschmelzen die Cluster im Datenraum (bzw. auf der SOM) und man erhält sowohl mit linearen als auch mit nichtlinearen Modellen sehr gute Klassifikationsergebnisse. Die Zuverlässigkeit der Methode kann also möglicherweise weiter gesteigert werden, indem der Datensatz dahingehend erweitert wird, dass für alle Stämme eine vergleichsweise hohe Diversität wie für *S. epidermidis* ATCC 35984 erreicht wird. Unter der Prämisse des maximal diversen Trainingsdatensatzes ist eine zuverlässige und schnelle Identifizierung von einzelnen Bakterienzellen ohne vorherige Kultivierung möglich. Dies wurde in der Arbeit abschließend überprüft, indem ein „verblindetes“ Testset mit den Methoden der paarweisen

MDA und SVMs untersucht wurde. Das Testset enthielt zusätzlich zu den Bakterienstämmen, die im Trainingsdatensatz enthalten sind, auch „unbekannte“ Bakterienstämme (d.h. Stämme die nicht im Trainingsdatensatz vorkommen). Die Erkennung von „unbekannten“ Spektren (Vorhersageausreißer) ist im industriellen „Online-Monitoring“ von ebenso großer Bedeutung wie die Klassifikation der Bakterien selbst, da gerade das Vorkommen von Bakterien, mit denen man im Reinraumumfeld nicht rechnet, ein alarmierendes Signal für Mängel im Herstellungsprozess sein kann. Für diesen Zweck erweist sich MDA erneut als eine zuverlässige Methode. Mit Hilfe der aus dem MDA Modell geschätzten Wahrscheinlichkeitsdichten können einfach und zuverlässig Vorhersageausreißer erkannt werden. Bei der Analyse des „verblindeten“ Testsets mittels MDA wurde bei einer Irrtumswahrscheinlichkeit von 20% der Großteil der enthaltenen unbekannten Bakterienstämme (84.0%) erkannt. Vor allem Bakterien, die von einer anderen Art stammen als die im Trainingsdatensatz enthaltenen, werden mit Hilfe der MDA sehr zuverlässig als Ausreißer detektiert. Eine befriedigende Ausreißererkennung mittels Ein-Klassen-SVM (engl. „One-Class Svm“) konnte dagegen mit dem bei der Klassifikation verwendeten RBF-Kernel nicht erreicht werden. Da der Erfolg der Ein-Klassen-SVM in hohem Maß von dem verwendeten Kernel sowie den eingesetzten Parametern abhängt, ist die Durchführung der Ausreißererkennung mit dieser Methode aufwändiger als unter Verwendung von MDA. Bei der Klassifikation des „verblindeten“ Testsets erreichte sowohl das SVM-Modell als auch die paarweise MDA eine sehr gute Vorhersagegenauigkeit. So wurden mit beiden Methoden annähernd alle Spektren richtig klassifiziert.

Zusammenfassend kann man festhalten, dass sich neben SVMs vor allem die paarweise MDA hervorragend für die Klassifikation von Bakterien mittels Mikro-Raman-Spektroskopie eignet. Neben einer mit den SVMs vergleichbaren Klassifikationsrate bietet die MDA zusätzlich die Möglichkeit zur Charakterisierung der vorhandenen Datenstruktur, was Rückschlüsse über die Zuverlässigkeit des Klassifikationserfolgs zulässt. Außerdem ist eine einfache Durchführung der Detektion von Vorhersageausreißern möglich, was für SVMs in dieser Arbeit zu keinen zufriedenstellenden Ergebnissen führte. Die Zuverlässigkeit der Klassifikation kann bei beiden Methoden durch *a posteriori* Wahrscheinlichkeiten eingeschätzt werden.

Das in dieser Arbeit beschriebene Verfahren zur Identifizierung von Bakterien mittels konfokaler Mikro-Raman-Spektroskopie und multivariater Datenanalyse stellt eine im Hinblick auf Zeitbedarf und Präzision sehr gute Methode zur Differenzierung von Mikroorganismen dar. Insbesondere die kurze Analysendauer und die Möglichkeit auch schwer kultivierbare bzw. langsam wachsende Organismen schnell und zuverlässig zu bestimmen, ist ein großer Vorteil der Methode.

Für die reibungslose Umsetzung in der Praxis, ist es notwendig, dass das Verfahren vollautomatisch abläuft. Ein für diesen Zweck geeignetes Gerätesetup wurde in [14] vorgestellt. Dabei wird in zwei Schritten vorgegangen. Zunächst erfolgt eine automatisierte auf Fluoreszenzbildgebung basierende Unterscheidung zwischen organischem und anorganischem Material. Im zweiten Schritt werden die potentiellen mikrobiellen Partikel mit Hilfe der Mikro-Raman-Spektroskopie sowie mit datenanalytischen Verfahren identifiziert.

In der Zukunft gilt es, die in dieser Arbeit erhaltenen, vielversprechenden Ergebnisse in der Praxis umzusetzen. Durch verschiedene Maßnahmen wie die Vergrößerung des Datensatzes und dessen Diversität kann die Methode eventuell noch verbessert werden. Außerdem bietet die „Gaussian Mixture“ Diskriminanzanalyse (MDA), die sich für die Klassifikation von Bakterien mittels Mikro-Raman-Spektroskopie als sehr nützlich erweist, zahlreiche Möglichkeiten zur Modifikation. Für diesen Zweck erweist sich die Toolbox MCLUST, die von Fraley and Raftery für die statistische Programiersprache R entwickelt wurde, als geeigneter Ausgangspunkt. Die Toolbox, die unter [144] erhältlich ist, bietet eine Vielzahl an Möglichkeiten für die Bildung von Gauss´schen Mischmodellen („Gaussian Mixtures“) mittels EM-Algorithmus (engl. Expectation-Maximization Algorithm) für die Clusteranalyse und für die Klassifikation. Die Toolbox basiert auf der Publikation von Fraley et al. [68], in der eine generalisierte Form der MDA (MclustDA) vorgeschlagen wurde. Bei der MclustDA ist es möglich eine separate Varianz-Kovarianz Matrix für jede Klasse bzw. Unterklasse zu bilden. Außerdem kann die Anzahl der Unterklassen sowohl zwischen als auch innerhalb der Klassen variieren. Auf diese Weise könnten die Klassifikationergebnisse dieser Studie eventuell weiter verbessert werden. Ein Nachteil dieser Methode ist die größere Anzahl an zu definierenden “Tuning”-Parametern, was das Risiko des “Overfittings” durch Modellselektion erhöht.

Neben der Anwendung im industriellen „Online-Monitoring“ von Reinräumen besteht die Möglichkeit, das entwickelte Verfahren auf andere Bakterienstämme und Anwendungsgebiete zu übertragen. Neben einer Vielzahl von Anwendungsmöglichkeiten besteht besonders im klinischen Umfeld ein großer Bedarf an einer schnellen und zuverlässigen Differenzierung von Mikroorganismen auf Stammebene.

5 Summary

In the pharmaceutical clean-room production as well as in the food-processing industry the production environment is controlled and maintained using very stringent protocols and guidelines to prevent microbial contamination of drugs and food. For this reason, there is a great interest in efficient technologies for the rapid identification of microorganisms, which can save production time and costs. In the past decade, various new techniques were developed, in order to replace conventional, time-consuming microbiological methods by more rapid approaches. Among these, vibrational spectroscopy has emerged as a very promising tool. By the invention of the confocal micro-Raman spectroscopy even single bacterial cells can be analyzed without previous cultivation step. This poses new experimental and data analytical challenges. This dissertation deals with data analytical aspects of the identification of bacteria by means of micro-Raman spectroscopy. A classification system was developed, which is suited to differentiate single bacterial cells using micro-Raman spectroscopic data. This technique is intended to be implemented in an automated in-process control system in industrial clean-room environments. For this purpose, a highly diverse dataset comprising 3642 micro-Raman spectra of 29 different strains of bacteria, which are commonly present in clean-rooms, was analyzed. In order to study the influence of different cultivation parameters on the classification success, the bacteria were cultivated under various conditions regarding cultivation medium, temperature and time.

The statistical analysis of the micro-Raman spectra consists of several phases. Apart from different pre-processing techniques including normalization and background correction methods and their combinations, linear and nonlinear classification algorithms were evaluated according to their ability to differentiate bacterial strains on the basis of their Raman spectra. The employed classification algorithms include partial least squares discriminant analysis (PLS-DA), linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), *k*-nearest neighbor classifier (*k*NN) and support vector machines (SVMs). In a first attempt the best prediction performance was achieved by a SVM model yielding 87.3% of correctly classified spectra. The prediction accuracy of some classifiers (PLS-DA,

LDA, MDA) could, however, be improved markedly by establishing multiple one-class-versus-one-class models. Predictions are then based on a major vote decision over all pairwise classifications. Using this pairwise approach the performance of MDA increased from 80.9% up to 86.6%, which is statistically not different from the performance of a support vector machine. In order to determine the existence of differences among several classification techniques, CVANOVA as well as a cross-validated Kruskal-Wallis test were performed.

In the end, the best classification rates for the differentiation of bacterial strains by means of micro-Raman spectroscopy were obtained by SVMs and pairwise MDA.
In the next step these two classification approaches were analyzed with respect to their robustness and overall usefulness for the given task.

Both classifiers (pairwise MDA and SVM) turned out to be stable regarding model selection and prediction accuracy when dealing with varying dataset splits. This was shown by performing two loops of 50-fold cross-validation. By the generation of multiple external test sets (outer loop) independent of model selection (inner loop) the absence of model selection bias could be shown. By using the double validation scheme with bootstrapping in the outer loop it could be shown that model selection and classification for both classifiers (pairwise MDA and SVM) are robust against the usage of smaller and more divers training datasets.

In a further step, MDA was utilized to illustrate the impact of different cultivation parameters on the dataset structure (i. e. data variability introduced owing to biological variability) and the classification performance. This knowledge is crucial for a reliable discrimination of microorganisms on single cell level, as microorganisms appear in different growth states in their natural environments. Therefore, a mere classification procedure without considering the data variability can bury sources of errors. It could be shown, that cultivation medium, temperature and time can be separated by the subcenters of MDA. This indicates that different cultivation parameters generate clusters in the variable space requiring classification approaches being able to recognize scattered class structures. By training a self organizing map (SOM) the clusters were visualized and the scattered clusters within single classes could be observed. Moreover, the congruity between cultivation parameters and clusters in the dataset could be visualized by the SOM. Therefore, for discriminating single bacteria without previous cultivation step by means of micro-Raman spectroscopy, classification approaches,

which are able to model flexible and scattered class structures, are superior to classifiers providing only linear decision boundaries. This, however, also indicates that the classification performance of approximately 87%, which is a quite promising result for the identification of bacteria on strain level, might not be transferred to the identification of microorganisms originating from an environment, which is not covered by the training dataset. Consequently, for the identification of bacterial strains without previous cultivation step a comprehensive spectral database, covering the natural variance of the microorganisms is necessary. In the present dataset, the strain *S. epidermidis* ATCC 35984 constitutes an example for covering maximum possible bacterial heterogeneity, since a great amount of bacteria grown under varying culturing parameters has been recorded. On a SOM it can be seen that the clusters of this strain become indistinct. The classification of this strain performed well by linear (94.8%) as well as by nonlinear approaches (99.1%). This indicates that the classification performance could possibly even be improved by enhancing the diversity of all bacterial strains included in the training dataset with respect to different cultivation conditions.

In the industrial framework, it may be important to assess the confidence level of the resulting predictions by *a posteriori* probabilities and it is indispensable to recognize prediction outliers, i.e. novel spectra based on microorganisms not belonging to the recorded dataset.

MDA allows a straightforward assessment of *a posteriori* probabilities. Recently, the estimation of *a posteriori* probabilities for support vector machines also became available. In order to obtain multi-class *a posteriori* probabilities for SVMs and pairwise MDA the two-class *a posteriori* probabilities can be coupled by using existing pairwise coupling techniques. In this study, it turned out that the classification performances based on the maximum multi-class *a posteriori* probabilities were equivalent to the classification performances obtained by a simple major vote scheme for both pairwise classifiers.

For the detection of prediction outliers, a one-class classification was performed for MDA and SVMs. For this purpose a “blinded” test set consisting of bacterial strains included in the training set as well as “unknown” bacterial strains were analyzed. Using MDA, it turned out that almost all bacterial strains (84.0%) not belonging to the training dataset (“novelties”) could be detected with an error of the first kind of 20%. The detection of novelties was

especially successful on species level (91.0%). The employment of the one-class SVM using an RBF-Kernel, yielded no satisfactory detection of the novelties included in the “blinded” testset. Using the latter the percentage of correctly identified novelties was as high as the percentage of false positives (spectra, which are detected as novelties, though they are not).
When classifying all spectra stemming from “known” bacterial strains of the “blinded” test set, for both classifiers almost all spectra were classified correctly.
Overall, the analysis demonstrates that SVMs as well as pairwise MDA are suited for a reliable and fast differentiation of single bacterial cells by means of micro-Raman spectroscopy. MDA additionally exhibits useful features for the differentiation of single bacteria by micro-Raman spectroscopy in terms of novelty detection, and interpretation of the model.
For the implementation of these techniques in an in-process control system in clean-rooms a fully automated process is necessary, which allows the investigation of a large number of particles in a very short time. A suitable setup (Online monitoring and identification of bioaerosol setup: OMIB setup) for this purpose was proposed by Rösch et al. [14]. They developed a laboratory instrument, in which the three detection methods microscopy, fluorescence microscopy and Raman spectroscopy are combined with data analytical tools and integrated in a fully automated system. With the help of fluorescence images biotic particles can be differentiated from abiotic particles. When a particle is detected as potentially relevant bio-particle, its position is determined and the confocal Raman measurements are performed. Subsequently, the Raman spectra of all detected bio-particles are analyzed with suitable data analytical tools.
In future research projects, the excellent results obtained by the developed classification system might be further improved by enhancing the diversity of the bacterial dataset. This can have beneficial effects on the classification success of linear as well as nonlinear classifiers. Additionally, the properties of MDA, which (beside SVMs) yielded the best classification results for the given task, should further be studied. For this purpose the toolbox MCLUST designed by Fraley and Raftery for the statistical language R, is a good starting point. The toolbox is available at [144] and offers a variety of tools for normal mixture modeling via expectation-maximization algorithm, model-based clustering, discriminant analysis and density estimation. In their work Fraley et al. extended MDA to a

generalized version of MDA called MclustDA, which allows the component covariance matrix and the number of subcenters to vary both within and between classes [68]. In this way, the obtained classification results might be further improved. A disadvantage of the generalized form of MDA is the accumulating number of parameters, which have to be adjusted by the user. This increases the risk of model selection bias. Furthermore the stability of the model can suffer from the estimation of separate variance-covariance matrices for each subclass.

The presented classification system, which is specially designed for the identification of bacteria in clean-room environments, can also be useful for other application fields. Especially within the clinical environment the rapid and reliable identificaiton of bacteria on strain level is of great interest.

Anhang

A Normal-Q-Q-Plots: Test auf Normalverteilung vor CVANOVA

In dieser Arbeit wird eine CVANOVA verwendet, um Unterschiede zwischen den Klassifikationsraten verschiedener Vorbehandlungs- und Klassifikationmethoden zu bewerten (siehe Kapitel 2.3.4.2 und 3.4.1.1.1). Im Vorfeld einer CVANOVA wird überprüft, ob die 50 Einzelvorhersagewerte aus der 50-fachen Kreuzvalidierung für die verschiedenen Methodenkombinationen die Kiterien der Normalverteilung und Varianzhomogenität erfüllen (siehe Kapitel 2.3.4.2.1).

Mit Hilfe von Normal-Quantil-Quantil-Plots (Normal-Q-Q-Plots) kann man grafisch überprüfen, ob eine empirische Verteilung (hier die Verteilung der Wiedererkennungsraten aus der 50-fachen Kreuzvalidierung) der angenommenen Normalverteilung entspricht. Dazu werden ausgewählte Quantile der beiden Verteilungen gegeneinander aufgetragen. In den folgenden Graphiken sind jeweils die Quantile der Standardnomalverteilung auf der x-Achse und die Quantile der empirischen Verteilung (Verteilung der Wiedererkennungsraten der 50-fachen Kreuzvalidierung) auf der y-Achse aufgetragen. Sind die Wiedererkennungsraten aus der 50-fachen Kreuzvalidierung normalverteilt, beschreiben die aufgetragenen Punkte annähernd eine Gerade. Starke Abweichungen von der Gerade weisen auf Abweichungen von der Normalverteilung hin.

A.1 Normal-Q-Q-Plots für PLS-DA-Ergebnisse

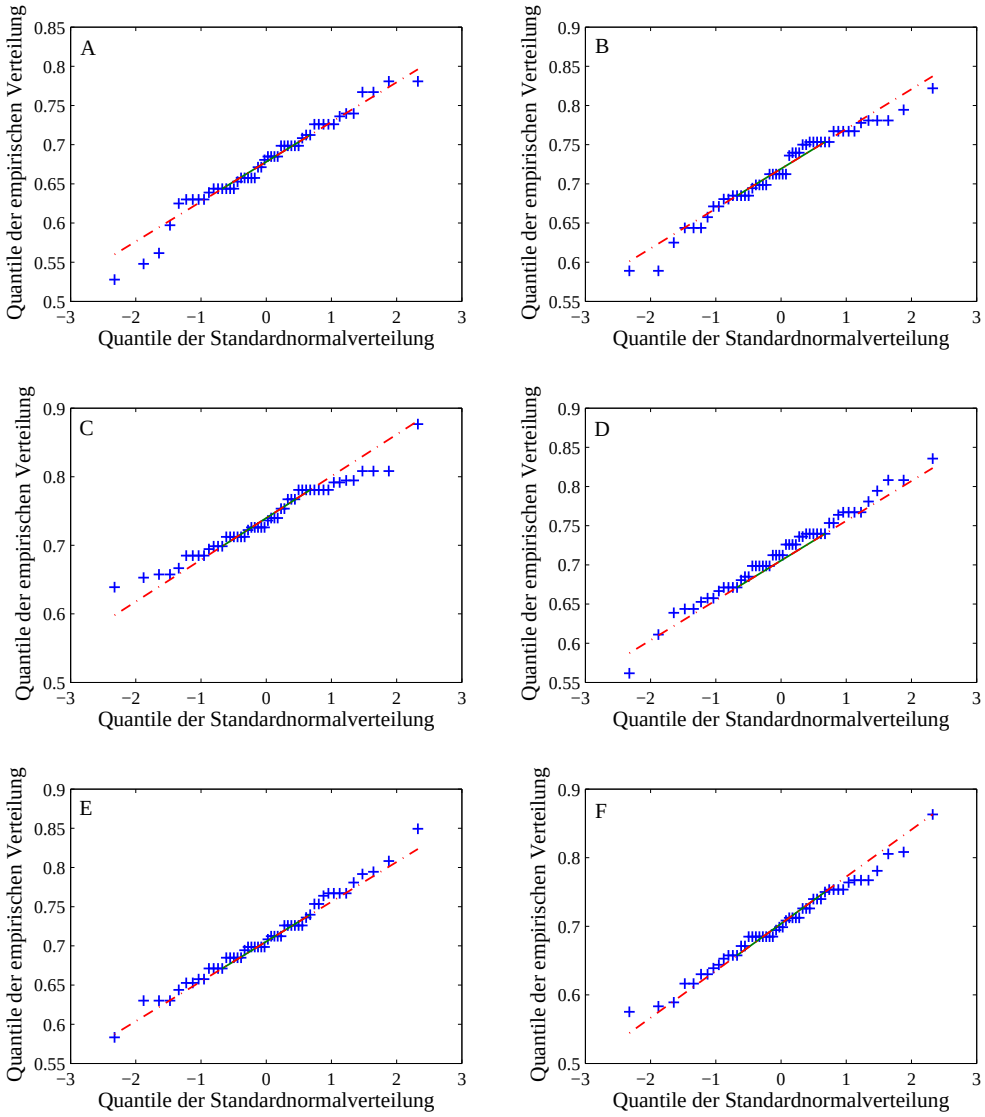


Abbildung A.1: Normal-Q-Q-Plots für die kreuzvalidierten Klassifikationraten von PLS-DA in Kombination mit mit verschiedenen Vorbehandlungsmethoden. Dabei sind die Quantile der Standardnormalverteilung (x-Achse) gegen die Quantile der empirischen Verteilung (Verteilung der Wiedererkennungsraten aus der Kreuzvalidierung) (y-Achse) aufgetragen. Verwendete Vorbehandlungsmethoden: A: INTERPOL, B: SPIKEELIM, C: VEKNORM, D: POLY4, E: 1.ABL, F: WHIT.

A.2 Normal-Q-Q-Plots für LDA-Ergebnisse

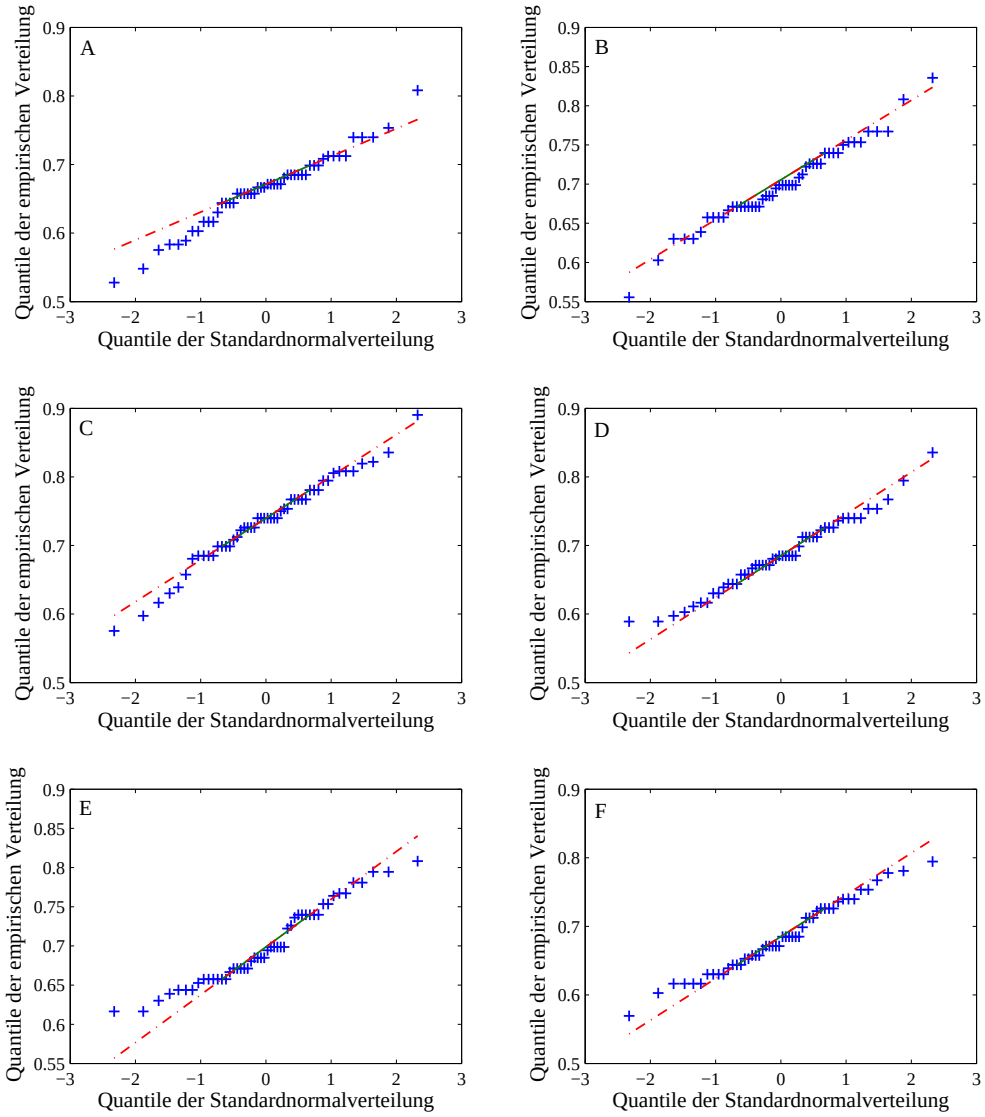


Abbildung A.2: Normal-Q-Q-Plots für die kreuzvalidierten Klassifikationraten von LDA in Kombination mit mit verschiedenen Vorbehandlungsmethoden. Dabei sind die Quantile der Standardnormalverteilung (x-Achse) gegen die Quantile der empirischen Verteilung (Verteilung der Wiedererkennungsraten aus der Kreuzvalidierung) (y-Achse) aufgetragen. Verwendete Vorbehandlungsmethoden: A: INTERPOL, B: SPIKEELIM, C: VEKNORM, D: POLY4, E: 1.ABL, F: WHIT.

A.3 Normal-Q-Q-Plots für QDA-Ergebnisse

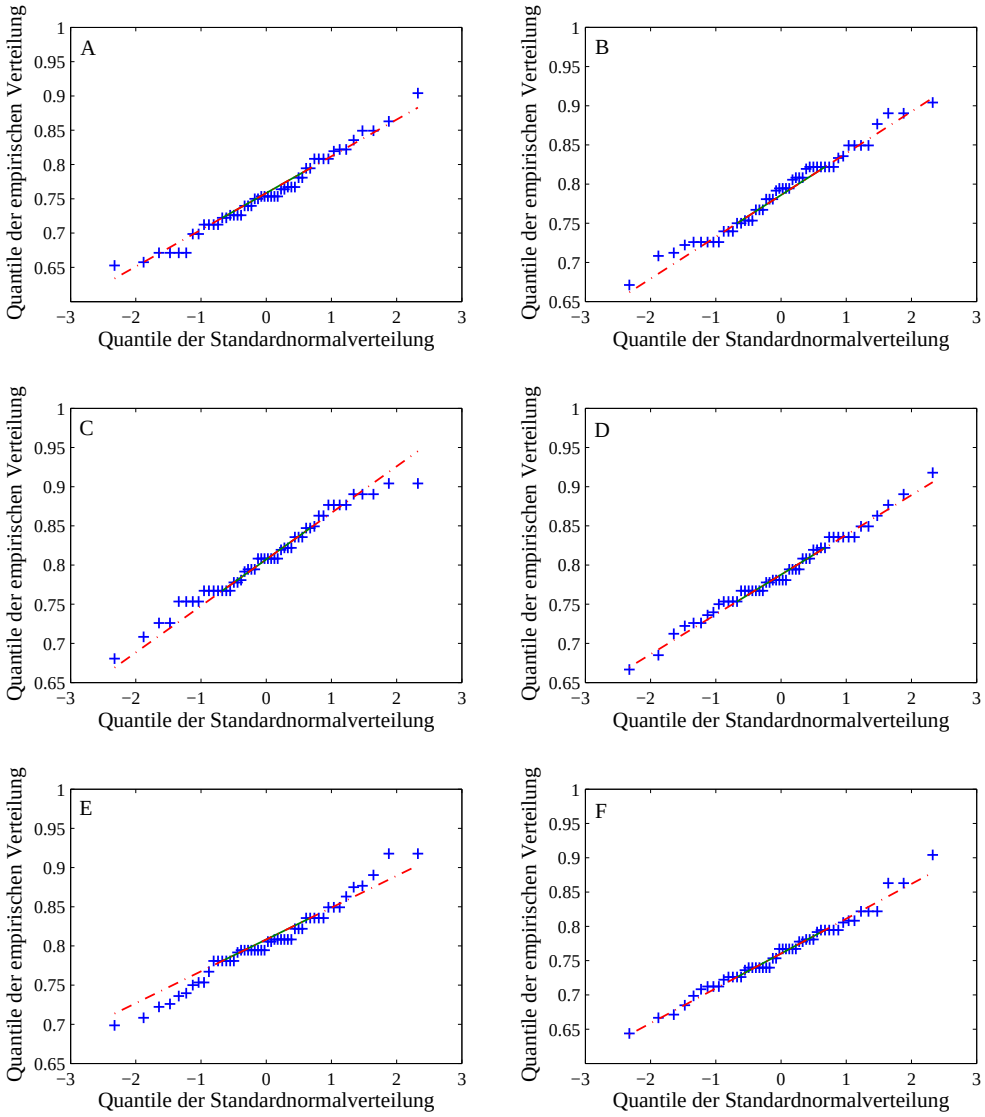


Abbildung A.3: Normal-Q-Q-Plots für die kreuzvalidierten Klassifikationraten von QDA in Kombination mit mit verschiedenen Vorbehandlungsmethoden. Dabei sind die Quantile der Standardnormalverteilung (x-Achse) gegen die Quantile der empirischen Verteilung (Verteilung der Wiedererkennungsraten aus der Kreuzvalidierung) (y-Achse) aufgetragen. Verwendete Vorbehandlungsmethoden: A: INTERPOL, B: SPIKEELIM, C: VEKNORM, D: POLY4, E: 1.ABL, F: WHIT.

A.4 Normal-Q-Q-Plots für MDA-Ergebnisse

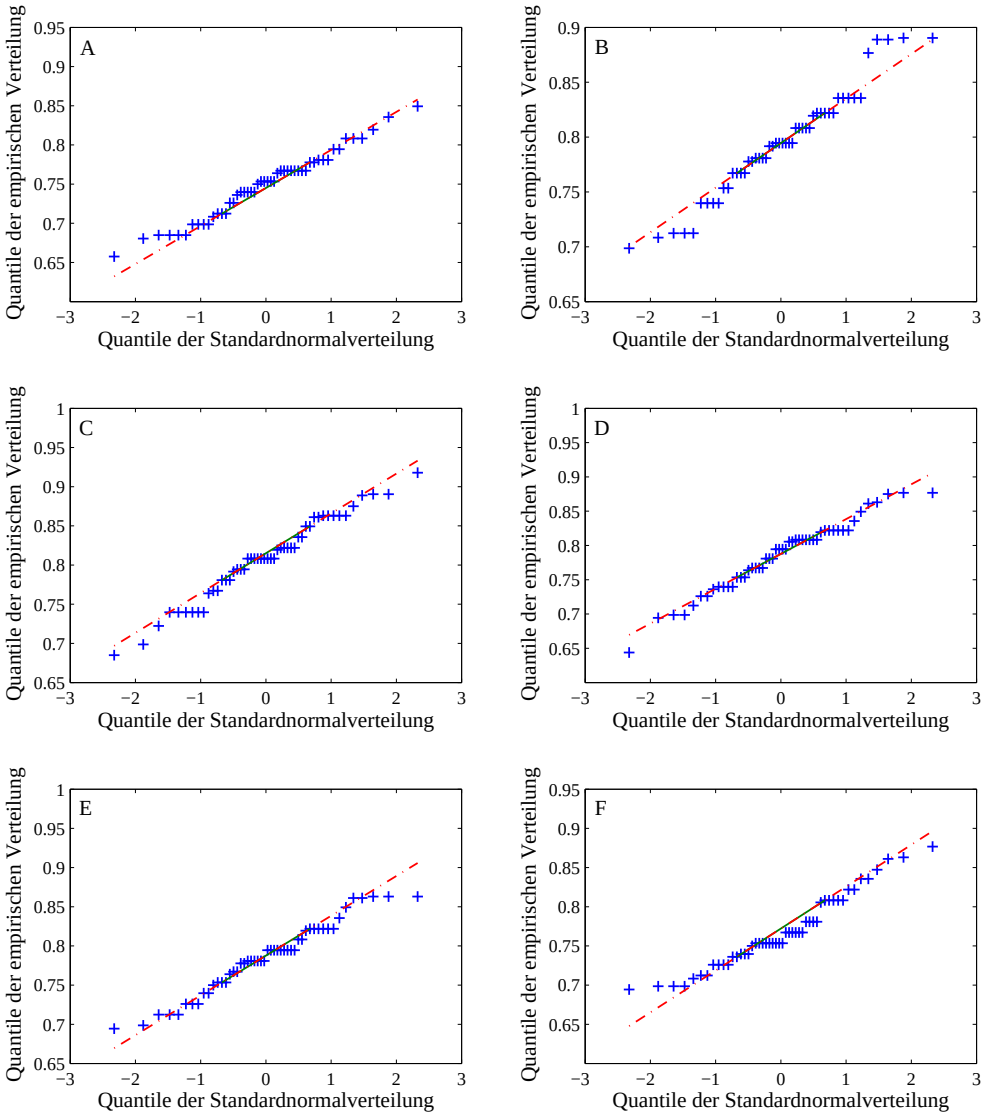


Abbildung A.4: Normal-Q-Q-Plots für die kreuzvalidierten Klassifikationraten von MDA in Kombination mit mit verschiedenen Vorbehandlungsmethoden. Dabei sind die Quantile der Standardnormalverteilung (x-Achse) gegen die Quantile der empirischen Verteilung (Verteilung der Wiedererkennungsraten aus der Kreuzvalidierung) (y-Achse) aufgetragen. Verwendete Vorbehandlungsmethoden: A: INTERPOL, B: SPIKEELIM, C: VEKNORM, D: POLY4, E: 1.ABL, F: WHIT.

A.5 Normal-Q-Q-Plots für *k*NN-Ergebnisse

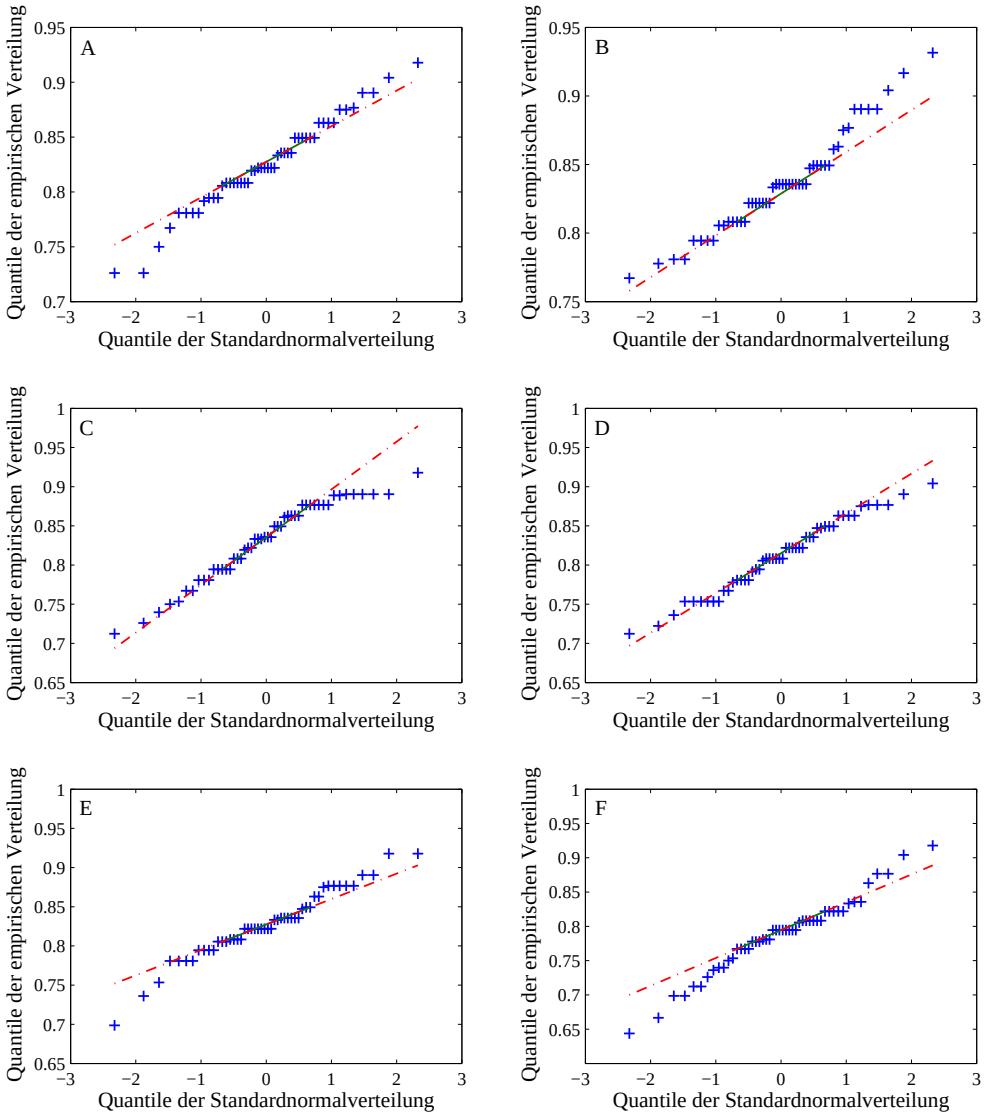


Abbildung A.5: Normal-Q-Q-Plots für die kreuzvalidierten Klassifikationraten von *k*NN in Kombination mit mit verschiedenen Vorbehandlungsmethoden. Dabei sind die Quantile der Standardnormalverteilung (x-Achse) gegen die Quantile der empirischen Verteilung (Verteilung der Wiedererkennungsraten aus der Kreuzvalidierung) (y-Achse) aufgetragen. Verwendete Vorbehandlungsmethoden: A: INTERPOL, B: SPIKEELIM, C: VEKNORM, D: POLY4, E: 1.ABL, F: WHIT.

A.6 Normal-Q-Q-Plots für SVM-Ergebnisse

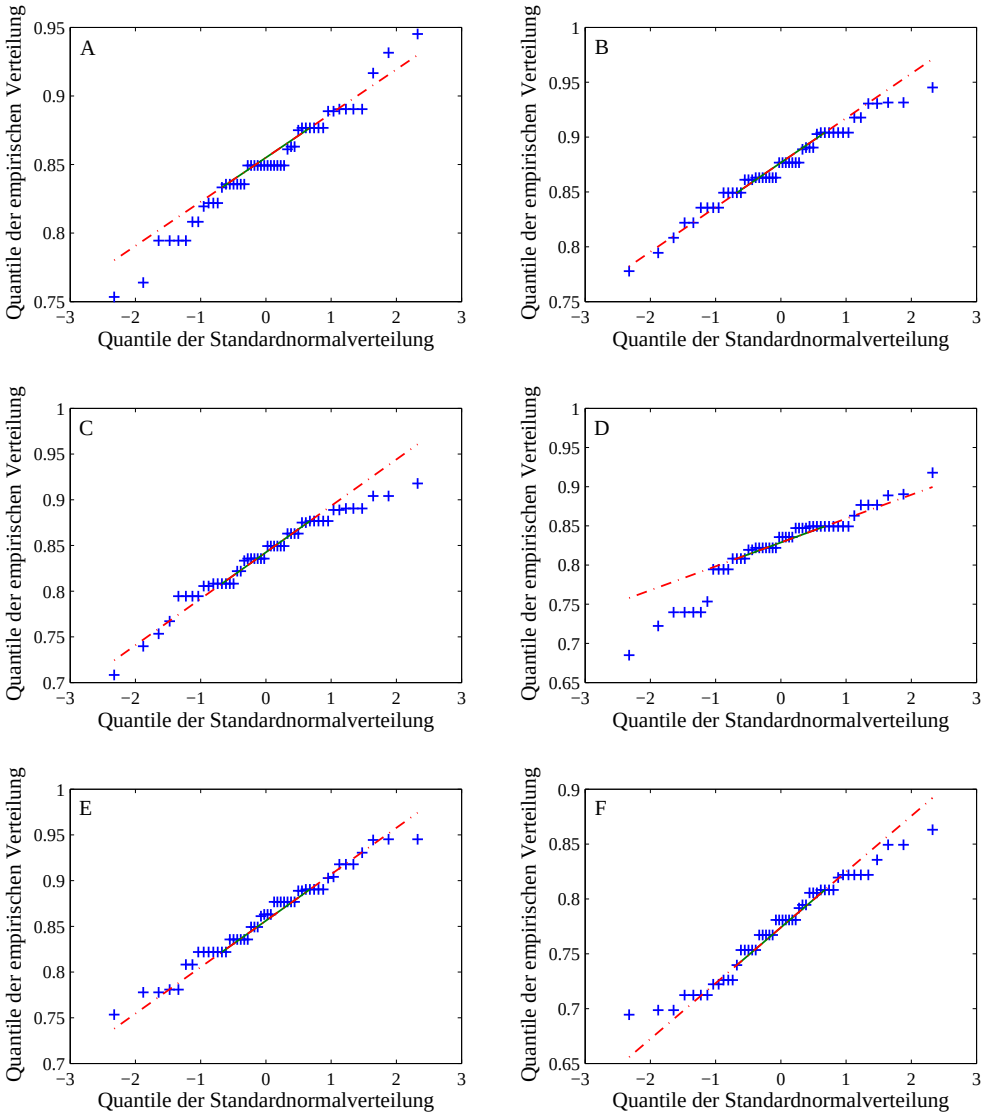


Abbildung A.6: Normal-Q-Q-Plots für die kreuzvalidierten Klassifikationraten von SVM in Kombination mit mit verschiedenen Vorbehandlungsmethoden. Dabei sind die Quantile der Standardnormalverteilung (x-Achse) gegen die Quantile der empirischen Verteilung (Verteilung der Wiedererkennungsraten aus der Kreuzvalidierung) (y-Achse) aufgetragen. Verwendete Vorbehandlungsmethoden: A: INTERPOL, B: SPIKEELIM, C: VEKNORM, D: POLY4, E: 1.ABL, F: WHIT.

B MATLAB Quellcode

Im Folgenden sind die in MATLAB programmierten Funktionen der „Gaussian Mixtures“ Diskriminanzanalyse (MDA) beschrieben, die in dieser Arbeit verwendet wurden.

B.1 *GaussianMix* (EM-Algorithmus)

Die zentrale Matlab-Funktion heißt *GaussianMix*. Sie enthält den in Kapitel 2.3.3.4.4 beschriebenen EM-Algorithmus und berechnet basierend auf den Trainingsdaten die geschätzten Modellparameter der MDA.

```
function [Psi, Centall, Covall, Mixprob] = GaussianMix (X, C, num_centers,
maxiter)
%[Psi, Centall, Covall, Mixprob] = GaussianMix (X, C, num_centers, maxiter)

% Input:        X: Data matrix of training objects,
%               rows = objects, columns = variables
%
%               C: Vector containing the class indices of the
%               training objects
%
%               num_centers: Number of subcenters for MDA.
%
%               maxiter: Maximum iterations of the EM-algorithm
%
% Output:       Psi: Matrix containing weigths corresponding
%               to the propabilities that the training objects belong
%               to one of the subcenters of MDA,
%               rows = objects, colums = subcenters of MDA
%
%               Centall: Matrix containing the centroids
%               of each subcenter of MDA,
%               rows = centroids, columns = variables
%
%               Covall: Pooled Variance-Covariance-Matrix
%               considering all subclasses and classes
%
```

```
%               Mixprob: Vector containing a priori probabilities
%               for each subcenter of MDA.
%
%
% Copyright:    Ulrike Schmid
%               University of Braunschweig, Institute
%               of technology, Department of
%               pharmaceutical chemistry, 2008



[m n] = size(X);
c = unique(C);
k = length(c);

%k-means clustering for initialization of parameters
[Psiinit Centallinit Covallinit Mixprobinit] = kmeansinit(X, C,
num_centers);

Centall = Centallinit;
Psilast = Psiinit;
Mixprob = Mixprobinit;
Psicurrent = zeros(m,k*num_centers);
tol = k*num_centers*eps;
Covall = Covallinit;

iter = 0;

%EM-Algorithm
while (sum(abs(Psilast-Psicurrent),2)>tol)& iter <= maxiter
    iter = iter+1;

    %Expectation Step
    if any(any(isnan(Covall)))
        Covall = Covallinit;
        [U S V] = svd(Covall);
        if S > eps;
            Covall = Covall;
```

```
    else
        Covall = Covall + diag(repmat(0.0000008,n,1));
    end
else
    [U S V] = svd(Covall);
    if S > eps;
        Covall = Covall;
    else
        Covall = Covall + diag(repmat(0.0000008,n,1));
    end
end

Psilast = Psicurrent;
Psii = zeros(m,k*num_centers);
Psicurrent = zeros(m,k*num_centers);
InvCo = inv(Covall);
for i = 1:k
    ii = find(C==c(i));
    for oo = (i*num_centers-num_centers+1):i*num_centers
        XOpt = X(ii,:)-repmat(Centall(oo,:),length(ii),1);
        Opt = -XOpt*InvCo;
        Opti = sum(Opt.*XOpt,2)./2;
        for o = 1:length(ii)
            Psii(ii(o),oo) = Mixprob(oo)*(exp(Opti(o)));
        end
    end
end

Probsum = sum(Psii,2);

for i = 1:k
    ii = find(C == c(i));
    for o = 1:length(ii)
        for oo = (i*num_centers-num_centers+1):i*num_centers
            if Probsum(ii(o))<=eps
                Psicurrent(ii(o),oo) = Psiinit(ii(o),oo);
            else
```

```
                    Psicurrent(ii(o),oo) = Psii(ii(o),oo)/Probsum(ii(o));
                end
            end
        end
    end

    %Maximization Step

    % Estimation of current mixing probabilities p(1,k*num_centers)
    Mixprob = sum(Psicurrent);
    i = 1;
    t = 0;

    while i < k*num_centers
        t = t+1;
        Mixprob(i:(i+num_centers-1)) = Mixprob(i:(i+num_centers-
1))./length(find(C == c(t)));
        i = i+num_centers;
    end

    % Estimation of current subclass centroids Centall (k*num_centers,n)
    for i = 1:k
        ii = find(C == c(i));
        for oo = (i*num_centers-num_centers+1):i*num_centers
            Centmix = X(ii,:).* repmat(Psicurrent(ii,oo),1,n);
            if sum(Psicurrent(ii,oo))<eps
                Centall(oo,:) = Centallinit(oo,:);
            else
                Centall(oo,:)= sum(Centmix)./sum(Psicurrent(ii,oo));
            end
        end
    end

    % Estimation of current covariance matrix(n,n)
    Cov = [];
    Covall = [];
```

```
    for i = 1:k
        ii = find(C==c(i));

        z = 0;
        Cova = zeros(num_centers, n,length(ii));

        for oo = (i*num_centers-num_centers+1):i*num_centers
            z = z+1;
            Cova(z,:,:) = (repmat(Psicurrent(ii,oo),1,n).*(X(ii,:)-
repmat(Centall(oo,:),length(ii),1)))';
        end

        for o = 1:length(ii)
            Covall(ii(o),:) = sum (Cova(:,:,o));
        end
    end
    Covall = (Covall'*Covall)./m;
    if any(any(isnan(Covall)))
        Covall = Covallinit;
    else
        [U S V] = svd(Covall);
        if S > eps;
            Covall = Covall;
        else
            Covall = Covall + diag(repmat(0.0000008,n,1));
        end
    end
end

Psi = Psicurrent;
```

B.2 *kmeansinit*

Innerhalb der Funktion *GaussianMix* wird die Funktion *kmeansinit* aufgerufen, die die Initialisierung der Modellparamter mittels *k*-Means Algorithmus vornimmt. Ein- und Ausgabevariablen entsprechen zum großen Teil den Ein- und Ausgabevariablen der Funktion *GaussianMix*. Da es sich bei den Ausgabeparametern um die Parameter handelt, mit denen der EM-Algorithmus initialisiert wird, enden die Parameterbezeichnungen mit „init“.

```
function [Psiinit, Centallinit, Covallinit, Mixprobinit] = kmeansinit(X, C,
num_centers)
%[Psiinit, Centallinit, Covallinit, Mixprobinit] = kmeansinit(X, C,
num_centers)

% Input:        X: Data matrix of training objects,
%               rows = objects, columns = variables
%
%               C: Vector containing the class indices of the
%               training objects
%
%               num_centers: Number of subcenters for MDA.
%
%               maxiter: Maximum iterations of the EM-algorithm
%
% Output:       Psiinit: Matrix containing weigths corresponding
%               to the propabilities that the training objects belong
%               to one of the subcenters of MDA,
%               rows = objects, colums = subcenters of MDA
%
%               Centallinit: Matrix containing the centroids
%               of each subcenter of MDA,
%               rows = centroids, columns = variables
%
%               Covallinit: Pooled Variance-Covariance-Matrix
%               considering all subclasses and classes
%
%               Mixprobinit: Vector containing a priori probabilities
%               for each subcenter of MDA.
```

```
%
%
% Copyright:    Ulrike Schmid
%               University of Braunschweig, Institute
%               of technology, Department of
%               pharmaceutical chemistry, 2008

[m n] = size(X);
c = unique(C);
k = length(c);
Centallinit = [];
Psiinit = [];

for i = 1:k
    psi = zeros(m,num_centers);
    [Cent Idx] = kmeans(X(C == c(i),:),num_centers);
    Centallinit = [Centallinit Cent'];
    if num_centers ~= 1
        for ii = 1:num_centers
            psi(C == c(i), ii) = (Idx == ii);
        end
    else
        psi(C == c(i),1) = 1;
    end
    Psiinit = [Psiinit psi];
end

Centallinit = Centallinit';

for i = 1:k
    ii = find(C==c(i));
    for o = 1:length(ii)
        z = 0;
        Cova = zeros(num_centers, n);
        for oo = (i*num_centers-num_centers+1):i*num_centers
            z = z+1;
```

```
            Cova(z,:) = Psiinit(ii(o),oo).*(X(ii(o),:) -
Centallinit(oo,:));
        end
        if num_centers ~=1
            Covallinit(ii(o),:) = sum (Cova);
        else
            Covallinit(ii(o),:) = Cova;
        end
    end
end
Covallinit = (Covallinit'*Covallinit)./m;

Mixprobinit = sum(Psiinit);
i = 1;
t=0;
while i < k*num_centers
    t = t+1;
    Mixprobinit(i:(i+num_centers-1)) = Mixprobinit(i:(i+num_centers-
1))./length(find(C == c(t)));
    i = i+num_centers;
end
```

B.3 *kmeans*

Innerhalb der Funktion *kmeansinit* wird die Funktion *kmeans* aufgerufen, die den eigentlichen *k*-Means Algorithmus (siehe Algorithmus 2.3) enthält.

```
function [Cent, Idx] = kmeans(X, num_centers)
%[Cent, Idx] = kmeans(X, num_centers)

% Input:        X: Data matrix of training objects,
%               rows = objects, columns = variables
%
%               num_centers: Number of clusters
%
% Output:       Cent: Matrix containing the centroids of each cluster
%               rows = centroids, columns = variables
```

```
%
%               Idx: Vector containing the cluster indices
%
%
% Copyright:    Ulrike Schmid
%               University of Braunschweig, Institute
%               of technology, Department of
%               pharmaceutical chemistry, 2008


k = num_centers;
[m n] = size(X);

if m < k
    error('Number of centroids larger than objects in training data');
end
thisSeed = 0;

rand('state', thisSeed);
RandState = rand('state');

v = randperm(size(X,1));
c = v(1:k);
Cent = X(c,:);
control = ones(1,m);
Idx = zeros(1,m);
opt = zeros(m,n,k);
while Idx ~= control
    control = Idx;
    for i=1:k
        opt = (repmat(Cent(i,:),m,1)-X).^2;
        D(i,:)=sqrt(sum(opt,2));
    end
    if num_centers ~=1
        [x Idx] = min(D);
    end
    for i = 1:k
```

```
        if any(Idx == i)
            Cent(i, :) = mean(X(Idx == i,:));
        else
            Cent(i,:) = mean(X);
        end
    end
end
Cent;
Idx = Idx';
```

B.4 *GaussianMixTest*

Die Funktion *GaussianMixTest* dient der Vorhersage neuer Testobjekte mit Hilfe des erstellten MDA-Modells.

```
function [CVal] = GaussianMixTest(X, XTest, C, mode, num_centers, dim)
%[CVal] = GaussianMixTest (X, XTest, C, mode, num_centers, dim)
%C must be a column vector

% Input:        X: Data matrix of training objects,
%               rows = objects, columns = variables
%
%               XTest: Data matrix of test objects,
%               rows = objects, columns = variables
%
%               C: Vector containing the class indices
%               of the training objects
%
%               mode: Defines the mode of principal component analysis
%               'u': The column vectors of matrix 'U' derived from
%                    singular value decomposition ([U, S, V] = svd(X));
%                    are used for classification by MDA
%               't': The scores (T) are used for classification
%               otherwise: Classification without preceeding
%                          principal component analysis is performed
%
```

```
%               num_centers: Number of subcenters for MDA
%
%               dim: Number of principal components
%
% Output:       CVal: Vector containing the predicted
%               class indices for all test objects
%
%
% Copyright:    Ulrike Schmid
%               University of Braunschweig, Institute
%               of technology, Department of
%               pharmaceutical chemistry, 2008



[m, n] = size(X);
CTrain = C;
c = unique(C);
k = length(c);

for i = 1:k
    Prior(i) = length(find(C==i))/m;
end

mTest = size(XTest, 1);
Xall=[X' XTest']';
[o oo] = size(Xall);

% Principal component analysis
switch lower(mode)
    case 'u'
        if o >= oo
            [U, S, V] = svd(Xall - ones(size(Xall, 1), 1) * mean(Xall), 0);
        else
            [V, S, U] = svd((Xall - ones(size(Xall, 1), 1) * mean(Xall))',
0);
        end
        XL = U(:, 1:dim);
```

```
    case 't'
        if o >= oo
            [U, S, V] = svd(Xall - ones(size(Xall, 1), 1) * mean(Xall), 0);
        else
            [V, S, U] = svd((Xall - ones(size(Xall, 1), 1) * mean(Xall))',
0);
        end
        XL = U(:, 1:dim) * S(1:dim, 1:dim);
    otherwise
        XL = Xall - ones(size(Xall, 1), 1) * mean(Xall);
end
XTrain = XL(1:m,:);
XTest = XL((m+1):end,:);

[m, n] = size(XTrain);
[mTest nTest] = size(XTest);
CVal = [];

%Classification
[Psi Centall Covall Mixprob] = GaussianMix (XTrain, CTrain,
num_centers,1000);

Posterior = [];
Covin = inv(Covall);
for i = 1:k
    Post = zeros(mTest,num_centers);
    h=0;
    for oo = (i*num_centers-num_centers+1):i*num_centers
        h=h+1;
        XTestopt = XTest - repmat(Centall(oo,:),mTest,1);
        Opt = -XTestopt*Covin;
        Opti = sum(Opt.*XTestopt,2)./2;
        for o = 1:mTest
            Post(o,h) = Mixprob(oo)*(exp(Opti(o)));
        end
    end
    Posterior(:,i) = Prior(i)*sum(Post,2);
```

```
end
[P class] = max(Posterior');
CVal = class;
```

B.5 *PairVoteMixTest*

Die Funktion *PairVoteMixTest* dient der Vorhersage neuer Testobjekte mit Hilfe des erstellten paarweisen MDA-Modells („major voting“, siehe Kapitel 2.3.3.5).

```
function [CVal] = PairVoteMixTest (X, XTest, C, mode, num_centers, dim)
%[CVal] = PairVoteMixTest (X, XTest, C, mode, num_centers, dim)
%C must be a column vector

% Input:        X: Data matrix of training objects,
%               rows = objects, columns = variables
%
%               XTest: Data matrix of test objects,
%               rows = objects, columns = variables
%
%               C: Vector containing the class indices
%               of the training objects
%
%               mode: Defines the mode of principal component analysis
%               'u': The column vectors of matrix 'U' derived from
%                    singular value decomposition ([U, S, V] = svd(X));
%                    are used for classification by MDA
%               't': The scores (T) are used for classification
%               otherwise: Classification without preceeding
%                          principal component analysis is performed
%
%               num_centers: Number of subcenters for MDA
%
%               dim: Number of principal components
%
% Output:       CVal: Vector containing the predicted
%               class indices for all test objects
%
```

```
%
% Copyright:    Ulrike Schmid
%               University of Braunschweig, Institute
%               of technology, Department of
%               pharmaceutical chemistry, 2008

[m, n] = size(X);
CTrain = C;
c = unique(C);
k = length(c);
mValid = size(XTest, 1);
Xall=[X' XTest']';
[o oo] = size(Xall);



%Principal component analysis
switch lower(mode)
    case 'u'
        if o >= oo
            [U, S, V] = svd(Xall - ones(size(Xall, 1), 1) * mean(Xall), 0);
        else
            [V, S, U] = svd((Xall - ones(size(Xall, 1), 1) * mean(Xall))',
0);
        end
        XL = U(:, 1:dim);
    case 't'
        if o >= oo
            [U, S, V] = svd(Xall - ones(size(Xall, 1), 1) * mean(Xall), 0);
        else
            [V, S, U] = svd((Xall - ones(size(Xall, 1), 1) * mean(Xall))',
0);
        end
        XL = U(:, 1:dim) * S(1:dim, 1:dim);
    otherwise
        XL = Xall - ones(size(Xall, 1), 1) * mean(Xall);
end
```

```
XTrain = XL(1:m,:);
XTest = XL((m+1):end,:);

[m, n] = size(XTrain);
CVal = [];

%pairwise classification
r = zeros(k,k);
for ii = 1:k-1
    for jj = ii+1:k
        hh = find(CTrain == c(ii)|CTrain == c(jj));
        XTrainp = XTrain(hh,:);
        CTrainp = CTrain(hh);

        Prior(1) = length(find(C==ii))/length(hh);
        Prior(2) = 1-Prior(1);



        [Psi Centall Covall Mixprob] = GaussianMix (XTrainp, CTrainp,
num_centers,1000);

        Posterior = zeros(mValid,2);
        Covin = inv(Covall);

        for g = 1:2
            Post = zeros(mValid,num_centers);
            h=0;
            for oo = (g*num_centers-num_centers+1):g*num_centers
                h=h+1;
                XTestopt = XTest - repmat(Centall(oo,:),mValid,1);
                Opt = -XTestopt*Covin;
                Opti = sum(Opt.*XTestopt,2)./2;
                for o = 1:mValid
                    Post(o,h) = Mixprob(oo)*(exp(Opti(o)));
                end
            end
            Posterior(:,g) = Prior(g).*sum(Post,2);
```

```
        end
        [P class] = max(Posterior');
        cc = unique(CTrainp);
        for z = 1:mValid
            if class(z) == 1
                r(cc(1),cc(2),z)= 1;

            else
                r(cc(2),cc(1),z) = 1;
            end
        end
    end
end

p=[];
for d = 1:mValid
    p(:,d) = sum(r(:,:,d),2);
end

%major voting
[t z] = max(p);
CVal = [CVal z];
```

Literaturverzeichnis

[1] M. Wilson, Microbial Inhabitants of Humans: Their Ecology and Role in Health and Disease Cambridge University Press, 2005.

[2] BPI. Pharma Kodex-Richtlinien, Gesetze, Empfehlungen, Band I und II

[3] http://ec.europa.eu/enterprise/pharmaceuticals/eudralex/homev4.htm. Abgerufen am 30. 04. 2009.

[4] Verordnung (EG) Nr. 853/2004 des Europäischen Parlaments und des Rates mit spezifischen Hygienevorschriften für Lebensmittel tierischen Ursprungs, Frankfurt am Main, 29. April 2004

[5] E. Memmert, DIN Deutsches Institut für Normung e. V. (Hrsg.), Reinraumtechnik (Normen-Handbuch), Beuth Verlag GmbH, Berlin, 2008.

[6] C.L. Wilkins und J. O. Lay Jr. (Hrsg.), Identification of Microorganisms by Mass Spectrometry, Wiley, Hoboken, New Jersey, 2006.

[7] J.A. Higgins und A.F. Azad, Use of Polymerase Chain Reaction to Detect Bacteria in Arthropods: A Review., J. Med. Entomol., 32 (1995) S. 213-222.

[8] D. Ivnitski, I. Abdel-Hamid, P. Atanasov und E. Wilkins, Biosensors for Detection of Pathogenic Bacteria, Biosens. Bioelectron., 14 (1999) S. 599-624.

[9] K. Maquelin, C. Kirschner, L.-P. Choo-Smith, N. v. d. Braak, H. P. Endtz, D. Naumann und G. J. Puppels, Identification of Medically Relevant Microorganisms by Vibrational Spectroscopy, J. Microbiol. Meth., 51 (2002) S. 255-271.

[10] D. Naumann, Infrared Spectroscopy in Microbiology, In R. A. Meyers (Hrsg.), Encyclopedia of Analytical Chemistry, John Wiley & Sons Ltd., Chichester, 2000, S. 102-131.

[11] T. Udelhoven, D. Naumann und J. Schmitt, Development of a Hierarchical Classification Cystem with Artificial Neural Networks and FT-IR Spectra for the Identification of Bacteria, Appl. Spectrosc., 54 (2000) S. 1471-1479.

[12] N. A. Ngo-Thi, C. Kirschner und D. Naumann, Characterization and Identification of Microorganisms by FT-IR Microspectrometry, Journal of Molecular Structure, 661-662 (2003) S. 371-380.

[13] K. Maquelin, L.-P. Choo-Smith, T. v. Vreeswijk, H. P. Endtz, B. Smith, R. Bennett, H. A. Bruining und G. J. Puppels, Raman Spectroscopic Method for Identification of Clinically Relevant Microorganisms Growing on Solid Culture Medium, Anal. Chem., 72 (2000) S. 12-19.

[14] P. Rösch, M. Harz, M. Krause, R. Petry, K.D. Peschke, H. Burkhardt, O. Ronneberger, A. Schüle, G. Schmauz, R. Riesenberg, A. Wuttig, M. Lankers, S. Hofer, H. Thiele, H.W. Motzkus und J. Popp, Online Monitoring and Identification of Bioaerosol (OMIB), In J. Popp und M. Strehle (Hrsg.), Biophotonics: Vision for a Better Health Care, Wiley-VCH, Weinheim, 2006, S. 89-165.

[15] P. Rösch, M. Harz, K.-D. Peschke, O. Ronneberger, H. Burkhardt und J. Popp, Identification of Single Eucaryotic Cells with Micro-Raman Spectroscopy, Biopolymers, 82 (2006) S. 312-316.

[16] P. Rösch, M. Harz, K.-D. Peschke, O. Ronneberger, H. Burkhardt, A. Schuele, G. Schmauz, M. Lankers, S. Hofer, H. Thiele, H.-W. Motzkus und J. Popp, On-Line Monitoring and Identification of Bioaerosols, Anal. Chem., 78 (2006) S. 2163-2170.

[17] P. Rösch, M. Harz, M. Schmitt, K. D. Peschke, O. Ronneberger, H. Burkhardt, H. W. Motzkus, M. Lankers, S. Hofer, H. Thiele und J. Popp, Chemotaxonomic Identification of Single Bacteria by Micro-Raman Spectroscopy: Application to Clean-Room-Relevant Biological Contaminations, Appl. Environ. Microbiol., 71 (2005) S. 1626-1637.

[18] D. Naumann, Infrared Spectroscopy in Microbiology, In R. A. Meyers (Hrsg.), Encyclopedia of Analytical Chemistry, Biomedical Spectroscopy, John Wiley and Sons, Chichester, 2000, S. 102-131.

[19] D. Naumann, FT-Infrared and FT-Raman Spectroscopy in Biomedical Research, In Gremlich H.-U. und B. Yan (Hrsg.), Infrared and Raman Spectroscopy of Biological Materials, Marcel Dekker, New York, 2001, S. 323-378.

[20] G. J. Puppels, F. F. de Mul, C. Otto, J. Greve, M. Robert-Nicoud, D. J. Arndt-Jovin und T. M. Jovin, Studying Single Living Cells and Chromosomes by Confocal Raman Microspectroscopy, Nature, 347 (1990) S. 301-303.

[21] K. Kneipp, A. S. Haka, H. Kneipp, K. Badizadegan, N. Yoshizawa, C. Boone, K. E. Shafer-Peltier, J. T. Motz, R. R. Dasari und M. S. Feld, Surface-Enhanced Raman Spectroscopy in Single Living Cells Using Gold Nanoparticles, Appl. Spectrosc., 56 (2002) S. 150-154.

[22] S. Nie und S. R. Emory, Probing Single Molecules and Single Nanoparticles by Surface-Enhanced Raman Scattering, Science, 275 (1997) S. 1102-1106.

[23] R. M. Jarvis und R. Goodacre, Discrimination of Bacteria Using Surface-Enhanced Raman Spectroscopy, Anal. Chem., 76 (2004) S. 40-47.

[24] R. M. Stöckle, Y. D. Suh, V. Deckert und R. Zenobi, Nanoscale Chemical Analysis by Tip-Enhanced Raman Spectroscopy, Chem. Phys. Lett., 318 (2000) S. 131-136.

[25] D. Richards, R. G. Milner, F. Huang und F. Festy, Tip-Enhanced Raman Microscopy: Practicalities and Limitations, J. Raman Spectrosc., 34 (2003) S. 663-667.

[26] D. Hutsebaut, K. Maquelin, P. De Vos, P. Vandenabeele, L. Moens und G. J. Puppels, Effect of Culture Conditions on the Achievable Taxonomic Resolution of Raman Spectroscopy Disclosed by Three Bacillus Species, Anal. Chem., 76 (2004) S. 6274-6281.

[27] C. Xie, J. Mace, M. A. Dinno, Y. Q. Li, W. Tang, R. J. Newton und P. J. Gemperline, Identification of Single Bacterial Cells in Aqueous Solution Using Confocal Laser Tweezers Raman Spectroscopy, Anal. Chem., 77 (2005) S. 4390-4397.

[28] M. Harz, P. Rösch, K. D. Peschke, O. Ronneberger, H. Burkhardt und J. Popp, Micro-Raman Spectroscopic Identification of Bacterial Cells of the Genus Staphylococcus and Dependence on their Cultivation Conditions, The Analyst, 130 (2005) S. 1543-1550.

[29] J. C. Platt, Probabilistic outputs for support vector machines and comparisions to regularized likelihood methods In A.J. Smola, P. Bartlett, B. Schölkopf und D. Schuurmans (Hrsg.), Advances in Large Margin Classifiers, MIT Press, Cambridge, 2000, S. 61-74.

[30] B. Schölkopf, R. C. Williamson, A. J. Smola, J. Shawe-Taylor und J. C. Platt, Support Vector Method for Novelty Detection, In S. A. Solla, T. K. Leen und K.-R. Müller (Hrsg.), Advances in Neural Information Processing Systems 12, MIT Press, 2000, S. 582-588.

[31] M. T. Madigan, J. M. Martinko und J. Parker, Brock Biology of Microorganisms, 10. Ausgabe, Prentice Hall, 2003.

[32] E. Brechner, B. Dinkelaker und D. Dreesmann, Online-Kompaktlexikon der Biologie. Wissenschaft Online, Spektrum Akademischer Verlag, Heidelberg, 2001.

[33] C. Gram, Über die isolierte Färbung der Schizomyceten in Schnitt- und Trockenpräparaten, Fortschr. Med., 15 (1884) S. 185-189.

[34] U Jayasooriya und R. D. Jenkins, Introduction to Raman Spectroscopy, In D. L. Andrews und A. A. Demidov (Hrsg.), Laser Spectroscopy, 2. Auflage, Kluwer Academic/Plenum Publishers, New York, 2002, S. 77ff.

[35] M. Hesse, H. Meier und B. Zeeh, Spektroskopische Methoden in der organischen Chemie, Thieme, Stuttgart 2002.

[36] www.Raman.de. Abgerufen am 30. 04. 2009.

[37] J. W. Riddle, P. W. Kabler, B. A. Kenner, R. H. Bordner, S. W. Rockwood und H. J. R. Stevenson, Bacterial Identification by Infrared Spectrophotometry, J. Bacteriol., 72 (1956) S. 593-603.

[38] W. C. Hayes, E. H. Melvin, J. M. Locke, C. A. Glass und F. R. Senti, Certain Factors Affecting the Infrared Spectra of Selected Microorganisms, Appl. Microbiol., 6 (1958) S. 298-304.

[39] K. P. Norris, Infrared Spectroscopy and its Application to Microbiology, J. Hyg., 57 (1959) S. 326-345.

[40] R. A. Dalterio, W. H. Nelson, D. Britt, J. F. Sperry und F. J. Purcell, A Resonance Raman Microprobe Study of Chromobacteria in Water, Appl. Spectrosc., 40 (1986) S. 271-272.

[41] R. A. Dalterio, M. Baek, W. H. Nelson, D. Britt, J. F. Sperry und F. J. Purcell, The Resonance Raman Microprobe Detection of Single Bacterial Cells from a Chromobacterial Mixture, Appl. Spectrosc., 41 (1987) S. 241-244.

[42] K. Danzer, H. Hobert, C. Fischbacher und K.-U. Jagemann, Chemometrik: Grundlagen und Anwendungen, Springer-Verlag, Berlin, 2001.

[43] W. Kessler, Multivariate Datenanalyse für die Pharma-, Bio- und Prozessanalytik, Wiley-VCH, Weinheim, 2006.

[44] P. J. Huber, Robust Statistics, Wiley, New York, 1981.

[45] N. Draper und H. Smith, Applied Regression Analysis, Wiley, New York, 1981.

[46] G. R. Phillips und J. M. Harris, Polynomial Filters for Data Sets with Outlying or Missing Observations: Application to Charge-Coupled-Device-Detected Raman Spectra Contaminated by Cosmic Rays, Anal. Chem., 62 (1990) S. 2351-2357.

[47] A. Savitzky und M. Golay, Smoothing and Differentiation of Data by Simplified Least Squares Procedures, Anal. Chem., 36 (1964) S. 1627-1639.

[48] L. Sachs, Angewandte Statistik, 8. Auflage, Springer, Berlin, 1997.

[49] C. D. Meyer, Matrix Analysis and Applied Linear Algebra, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2000.

[50] V. Mazet, C. Carteret, D. Brie, J. Idier und B. Humbert, Background Removal from Spectra by Designing and Minimising a Non-Quadratic Cost Function, Chemom. Intell. Lab. Sys., 76 (2005) S. 121-133.

[51] P. Eilers, A Perfect Smoother, Anal. Chem., 75 (2003) S. 3631-3636.

[52] E.T. Whittaker, On a New Method of Graduation, Proceedings of Edinburgh Mathematical Society, 41 (1923) S. 63-75.

[53] I. T. Jolliffe, Principal Component Analysis, Springer, New York, 1986.

[54] M. E. Wall, A. Rechtsteiner und L. M. Rocha, Singular Value Decomposition and Principal Component Analysis, In D. P. Berrar, W. Dubitzky und M. Granzow (Hrsg.), A Practical Approach to Microarray Data Analysis, Kluwer, Norwell, MA, 2003, S. 91-109.

[55] W.C. Chang, On Using Principal Components before Separating a Mixture of two Multivariate Normal Distributions, J. Roy. Stat. Soc. C, Applied Statistics, 32 (1983) S. 267-275.

[56] H. Wold, in F. N. David (Hrsg.), Research Papers in Statistics, Wiley, 1966, S. 411-444.

[57] H. Wold, Model Construction and Evaluation When Theoretical Knowledge Is Scare: Theory and Application of Partial Least Squares In J. Kmenta und J. B. Ramsey (Hrsg.), Evaluation of Econometric Models, Academic Press, New York, 1980, S. 47-74.

[58] R. De Maesschalck, D. Jouan-Rimbaud und D.L. Massart, The Mahalanobis Distance, Chemom. Intell. Lab. Sys., 50 (2000) S. 1-18.

[59] R.G. Brereton, Consequences of Sample Size, Variable Selection, and Model Validation and Optimisation, for Predicting Classification Ability from Analytical Data, TrAC-Trends in Analytical Chemistry, 25 (2006) S. 1103-1111.

[60] T. Hastie, R. Tibshirani und J. Friedman, The Elements of Statistical Learning, Springer-Verlag, Berlin, 2001.

[61] http://neil.fraser.name/writing/tank/. Abgerufen am 30. 04. 2009.

[62] R. O. Duda, P. E. Hart und D. G. Stork, Pattern Classification, 2. Ausgabe, John Wiley & Sons, Inc., New York, 2001.

[63] M. Barker und W. Rayens, Partial Least Squares for Discrimination, J. Chemom., 17 (2003) S. 166-173.

[64] R. A. Fisher, The Use of Multiple Measurements in Taxonomic Problems, Ann. Eugenics, 7 (1936) S. 179-188.

[65] W. Wu, Y. Mallet, B. Walczak, W. Penninckx, D.L. Massart, S. Heuerding und F. Erni, Comparison of Regularized Discriminant Analysis, Linear Discriminant Analysis and Quadratic Discriminant Analysis, Applied to NIR-Data, Anal. Chim. Acta, 329 (1996) S. 257-265.

[66] T. Hastie und R.J. Tibshirani, Discriminant Analysis by Gaussian Mixtures, J. Roy. Statist. Soc. B, 58 (1996) S. 155-176.

[67] D. Ormoneit und Tresp V., Averaging, Maximum Penalized Likelihood and Bayesian Estimation for Improving Gaussian Mixture Probability Density Estimates, IEEE Trans. on Neural Networks, 9 (1998) S. 639-649.

[68] C. Fraley und A.E. Raftery, Model-Based Clustering, Discriminant Analysis, and Density Estimation, J. Amer. Statist. Soc., 97 (2002) S. 611-631.

[69] J. Aldrich, R. A. Fisher and the Making of Maximum Likelihood 1912-1922, Statist. Sci., 12 (1997) S. 162-176.

[70] C.M. Bishop, Novelty Detection and Neural Network Validation, IEE Conference on Vision, Image and Signal Processing, Proceedings of the Conference, 1994.

[71] L. Tarassenko, P. Hayton, N. Cerneaz und M. Brady, Novelty Detection for the Identification of Masses in Mammograms, 4 th IEE International Conference on Artificial Neural Networks, Proceedings of the Conference, Cambridge, 1995.

[72] D. G. Altman und J. M. Bland, Statistics Notes, Diagnostic Tests 1: Sensitivity and Specificity, BMJ, 308 (1994) S. 1552.

[73] M. Markou und S. Singh, Novelty Detection: A Review - Part 1: Statistical Approaches, Signal Process., 83 (2003) S. 2481-2497.

[74] J. L. Villa, R. Boqué und J. Ferré, Calculation of the Probability of Correct Classification in Probabilistic Bagged k-Nearest Neighbours, Chemom. Intell. Lab. Sys., 94 (2008) S. 51-59.

[75] C. C. Holmes und N. M. Adams, A Probabilistic Nearest Neighbour Method for Statistical Pattern Recognition, J. Roy. Statist. Soc. B, 64 (2002) S. 295-306.

[76] V.N. Vapnik, The Nature of Statistical Learning Theory, Springer-Verlag, Berlin, 1995.

[77] C. Cortes und V. N. Vapnik, Support Vector Networks, Mach. Learn., 20 (1995) S. 273-297.

[78] A. J. Smola und B. Schölkopf, A Tutorial on Support Vector Regression Stat. Comput., 14 (2004) S. 199-222.

[79] H.-T. Lin, C.-J. Lin und R. C. Weng, A Note on Platt´s Probabilistic Outputs for Support Vector Machines, Technical Report, Department of Computer Science, National Taiwan University, 2003

[80] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola und R. C. Williamson, Estimating the Support of a High-Dimensional distribution, Neural Computation, 13 (2001) S. 1443-1471.

[81] J. Friedman, Another Approach to Polychotomous Classification, Technical Report, Stanford University, 1996.

[82] C.-W. Hsu und C.-J. Lin, A Comparison of Methods for Multi-Class Support Vector Machines, IEEE Trans. on Neural Networks, 13 (2002) S. 415-425.

[83] T.-K. Huang, R. C. Weng und C.-J. Lin, Generalized Bradley-Terry Models and Multi-Class Probability Estimates, J. Mach. Learn. Res., 7 (2006) S. 85-115.

[84] S. Knerr, L. Personnaz und B. Dreyfus, Single-Layer Learning Revisited: A Stepwise Procedure for Building and Training a Neural Network, In F. Fogelman Soulié und J. Hérault (Hrsg.), Neurocomputing: Algorithms, Architectures and Applications, Springer-Verlag, Berlin, 1990, S. 41-50.

[85] T. Hastie und R. Tibshirani, Classification by Pairwise Coupling, Ann. Statist., 26 (1998) S. 451-471.

[86] R.-F. Wu und C.-J. Lin, Probability Estimates for Multi-Class Classification by Pairwise Coupling, J. Mach. Learn. Res., 5 (2004) S. 975-1005.

[87] D. Price, S. Knerr, L. Personnaz und G. Dreyfus, Pairwise neural network classifiers with probabilistic outputs, In G. Tesauro, D. Touretzky und T. Leen (Hrsg.), Neural Information Processing Systems, Volume 7, The MIT Press, 1995, S. 1109-1116.

[88] P. Refregier und F. Vallet, Probabilistic Approach for Multiclass Classification with Neural Networks, International Conference on Artificial Networks, Proceedings of the Conference, 1991.

[89] C.-C. Chang und C.-L. Lin, LIBSVM: A Library for Support Vector Machines, Software erhältlich unter http://www.csie.ntu.edu.tw/~cjlin/libsvm (2001).

[90] M. Moreira und E. Mayoraz, Improving Pairwise Coupling Classification with Error Correcting Classifiers, Tenth European Conference on Machine Learning, Proceedings of the Conference, 1998.

[91] Z. Li, S. Tang und S. Yan, Multi-Class SVM Classifier Based on Pairwise Coupling, In S.-W. Lee und Verro. A. (Hrsg.), Pattern Recognition with Support Vector Machines, Springer, Berlin / Heidelberg, 2002.

[92] D. M. Hawkins, S. C. Basak und D. Mills, Assessing Model Fit by Cross-Validation, J. Chem. Inf. Comput. Sci., 43 (2003) S. 579-586.

[93] A. Blum, A. Kalai und J. Langford, Beating the Hold-Out: Bounds for K-fold and Progressive Cross-Validation, Twelfth International Conference on Computational Learning Theory, Proceedings of the Conference, 1999.

[94] S. Wold, On the Use of Cross-Validation to Assess Performance in Multivariate Prediction, Technometrics, 20 (1978) S. 397-405.

[95] S. Geisser, The Predictive Sample Reuse Method with Applications, J. Amer. Statist. Soc., 70 (1975) S. 320-328.

[96] B. Efron und R. Tibshirani, An Introduction to the Bootstrap, Chapman & Hall / CRC, New York, 1994.

[97] B. Efron, Estimating the Error Rate of a Prediction Rule: Improvement on Cross-Validation, J. Amer. Statist. Soc., 78 (1983) S. 316-331.

[98] E. Anderssen, K. Dyrstad, F. Westad und H. Martens, Reducing Over-Optimism in Variable Selection by Cross-Model Validation, Chemom. Intell. Lab. Sys., 84 (2006) S. 69-74.

[99] S. J. Dixon und R. G. Brereton, Comparison of Performance of Five Common Classifiers Represented as Boundary Methods: Euclidean Distance to Centroids, Linear Discriminant Analysis, Quadratic Discriminant Analysis, Learning Vector Quantization and Support Vector Machines, as Dependent on Data Structure, Chemom. Intell. Lab. Sys., 95 (2009) S. 1-17.

[100] S. J. Dixon, Y Xu, R. G. Brereton, H. A. Soini, M. V Novotny, E. Oberzaucher, K. Grammer und D. J. Penn, Pattern Recognition of Gas Chromatography Mass Spectrometry of Human Volatiles in Sweat to Distinguish the Sex of Subjects and Determine Potential Discriminatory Marker Peaks, Chemom. Intell. Lab. Sys., 87 (2007) S. 161-172.

[101] P. A. Lachenbruch und M. Mickey, Estimation of Error Rates in Discriminant Analysis, Technometrics, 10 (1968) S. 1-11.

[102] M. Stone, Cross-Validatory Choice and Assessment of Statistical Predictions, J. Roy. Statist. Soc. B, 36 (1974) S. 111-147.

[103] P. Zhang, Model Selection via Multifold Cross Validation, Ann. Statist., 21 (1993) S. 299-313.

[104] L. Breiman und P. Spector, Submodel Selection and Evaluation in Regression: The X-Random Case, Int. Stat. Rev., 60 (1992) S. 291-319.

[105] K. Baumann, H. Albert und M. von Korff, A Systematic Evaluation of the Benefits and Hazards of Variable Selection in Latent Variable Regression. Part I. Search Algorithm, Theory and Simulations, J. Chemom., 16 (2002) S. 339-350.

[106] K. Baumann, H. Albert und M. von Korff, A Systematic Evaluation of the Benefits and Hazards of Variable Selection in Latent Variable Regression. Part II. Practical Applications, J. Chemom., 16 (2002) S. 351-360.

[107] R. Kohavi, A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection, International Joint Conference on Artificial Intelligence, Proceedings of the Conference, 1995.

[108] J. Shao, Linear Model Selection by Cross-Validation, J. Amer. Statist. Soc., 88 (1993) S. 486-494.

[109] A. M. Molinaro, R. Simon und R. M. Pfeiffer, Prediction Error Estimation: A Comparison of Resampling Methods, Bioinformatics, 21 (2005) S. 3301-3307.

[110] B. Efron und R. Tibshirani, Improvements on Cross-Validation: The .632+ Bootstrap Method, J. Amer. Statist. Soc., 92 (1997) S. 548-560.

[111] J. Shao und D. Tu, The Jackknife and Bootstrap, Springer-Verlag, New York, 1996, S. 309.

[112] A. Elisseeff und M. Pontil, Leave-One-Out Error and Stability of Learning Algorithms with Applications, Advances in Learning Theory: Methods, Models and Applications, Volume 190 of NATO Science Series III: Computer & Systems Sciences

[113] L. Breiman, J. Friedman, J. S. Stone und R. A. Olshen, Classification and Regression Trees, Chapman & Hall/CRC, 1984.

[114] P. Burman, A Comparative Study of Ordinary Cross-Validation, v-Fold Cross-Validation and the Repeated Learning-Testing Methods, Biometrika, 76 (1989) S. 503-514.

[115] B. Efron, Another Look at the Jackknife, Ann. Stat., 7 (1979) S. 1-26.

[116] J. Hartung und B. Elpelt, Multivariate Statistik: Lehr- und Handbuch der angewandten Statistik, Oldenbourg, 1999.

[117] U.G. Indahl und T. Naes, Evaluation of Alternative Spectral Feature Extraction Methods of Textural Images for Multivariate Modeling, J. Chemom., 12 (1998) S. 261-278.

[118] M. R. Harwell, E. N. Rubinstein, W. S. Hayes und C. C. Olds, Summarizing Monte Carlo Results in Methodological Research: The One- and Two-Factor Fixed Effects ANOVA Cases, J. Educ. Stat., 17 (1992) S. 315-339.

[119] G. V. Glass, P. D. Peckham und J. R. Sanders, Consequences of Failure to Meet Assumptions Underlying the Fixed Effects Analysis of Variance and Covariance, Rev. Educ. Res., 42 (1972) S. 237-288.

[120] R. R. Wilcox, How Many Discoveries Have Been Lost by Ignoring Modern Statistical Methods?, Am. Psychol., 53 (1998) S. 300-314.

[121] C.A. Boneau, The Effects of Violations of Assumptions Underlying the t-Test, Psychol. Bull., 57 (1960) S. 49-64.

[122] W. G. Cochran, Some Consequences when the Assumptions for the Analysis of Variance are not Satisfied, Biometrics, 3 (1947) S. 22-38.

[123] G. E. P. Box, Some Theorems on Quadratic Forms Applied in the Study of Analysis of Variance Problems, I. Effects of Inequality of Variance in the One-Way Classification, Ann. Math. Statist., 25 (1954) S. 290-302.

[124] B. J. Winer, D. R. Brown und K. M. Michels, Statistical Principles in Experimental Design, McGraw-Hill, New York, 1991.

[125] J. Hartung, B. Elpelt und K.-H. Klösener, Statistik, Oldenbourg Wissenschaftsverlag GmbH, München, 2002.

[126] G. E. P. Box, Non-Normality and Tests on Variances, Biometrika, 40 (1953) S. 318-335.

[127] C. X. Wells und J. M Hintze, Dealing with Assumptions Underlying Statistical Tests, Psychol. Schools, 44 (2007) S. 495-502.

[128] K. A. McGuinness, Of Rowing Boats, Ocean Liners and Tests of the ANOVA Homogeneity of Variance Assumption, Austral Ecology, 27 (2002) S. 681-688.

[129] M. S. Bartlett, Properties of Sufficiency and Statistical Tests, Proc. Roy. Statist. Soc. A, 160 (1937) S. 268-282.

[130] W. Köhler, G. Schachtel und P. Voleske, Biostatistik, 4. Auflage, Springer, 2007.

[131] H. van der Voet, Comparing the Predictive Accuracy of Models Using a Simple Randomization Test, Chemom. Intell. Lab. Sys., 25 (1994) S. 313-323.

[132] E. V. Thomas, Non-Parametric Statistical Methods for Multivariate Calibration Model Selection and Comparision, J. Chemom., 17 (2003) S. 653-659.

[133] H. R. Cederkvist, A. H. Aastveit und T. Naes, A Comparison of Methods for Testing Differences in Predictive Ability, J. Chemom., 19 (2005) S. 500-509.

[134] T. G. Dietterich, Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms, Neural Comput., 10 (1998) S. 1895-1924.

[135] Y. Hochberg und A. C. Tamhane, Multiple Comparison Procedures, Wiley, 1987.

[136] W. H. Kruskal und W. A. Wallis, Use of Ranks in One-Criterion Variance Analysis, J. Amer. Statist. Soc., 47 (1952) S. 583-621.

[137] K. Boehnke, F- and H-Test Assumptions Revisited, Educ. Psychol. Meas., 44 (1984) S. 609-617.

[138] B. E. Dom, An Information-Theoretic External Cluster-Validity Measure, IBM Research Report RJ 10219, 2001.

[139] T. Kohonen, Self-Organizing Maps, Springer-Verlag, Secaucus, New York, 1997.

[140] A. Zell, Simulation Neuronaler Netze, Addison-Wesley, Bonn, 1994.

[141] http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/#9, Abgerufen am 30. 04. 2009.

[142] N. K. Afseth, V. H. Segtnan und J. P. Wold, Raman Spectra of Biological Samples: A Study of Preprocessing Methods, Appl. Spectrosc., 60 (2006) S. 1358-1367.

[143] O. E. De Noord, The Influence of Data Preprocessing on the Robustness and Parsimony of Multivariate Calibration Models, Chemom. Intell. Lab. Sys., 23 (1994) S. 65-70.

[144] http://www.stat.washington.edu/mclust/. Abgerufen am 30. 04. 2009.

Danke!

Zunächst möchte ich ganz herzlich meinem Doktorvater Knut für die interessante, lehrreiche und schöne Zeit in seiner Arbeitsgruppe danken. Mit didaktischem Geschick hat er mich als „Neuling“ in die Welt der Chemometrik eingeführt und mich während meiner Promotionszeit mit gutem Rat und wertvollen Anregungen begleitet. Danke Knut!
Daneben geht ein großer Dank an Herrn Professor Hermann Wätzig für die Übernahme des Koreferats und die Bereitschaft als Prüfer in der Disputation zu fungieren. Ebenso danke ich Herrn Prof. Ingo Rustenbeck für seine Beteiligung an der Prüfungskommision.
Ein großes Dankeschön gilt auch unseren Kooperationspartnern am Institut für physikalische Chemie der Universität Jena unter Leitung von Professor Jürgen Popp, die durch die Überlassung ihrer Daten dieses spannende Projekt erst möglich gemacht haben. Besonders ist die gute und fruchtbare Zusammenarbeit mit Dr. Ute Neugebauer zu erwähnen.
Nicht zuletzt möchte ich allen Baumännern, Holzgräblern, Kunickals und Wätzigs für die schöne gemeinsame Zeit in Würzburg und Braunschweig danken. Ganz besonders geht dieser Dank an Markus Kossner, Joseph Scheiber, Sebastian Rohrer, meinen Bürokollegen Florian Kölling, Jan Dreher, meine Korrekturleserin Christina Anthes, Wiebke Brandt, Anja Becker, Renate Determann, meine ehemaligen Labornachbarn Hendrik Stukenbrock und Simone Schröder, Phillip Hasemann, Stephanie Ludewig und Magnus Matz.
Desweiteren bedanke ich mich herzlich bei Herrn Professor Conrad Kunick und den „Mitstreitern“ des 3. Semesters für die nette und lustige Arbeitsatmosphäre bei der Planung und Betreuung von Praktika, Kolloquien und Klausuren.
Vielen Dank auch an Frank Roese für sein technisches Know-how und seinen Humor sowie an Klaus Hartmann, Matthias Söchtig und Eduard Hinz für ihre immer kompetente, nette und hilfsbereite Unterstützung bei der Meisterung des Uni-Alltags.
Mein abschließender Dank gilt meinen Eltern, meinen Geschwistern und Markus, die durch ihre Geduld und Unterstützung einen großen Teil zum Gelingen dieser Arbeit beigetragen haben.

Lebenslauf

Ulrike Schmid, geboren am 19. 07. 1977 in Krumbach (Schwaben)

09/1983 bis 07/1988	Grundschule (Volksschule Pfaffenhausen)
09/1988 bis 06/1997	Gymnasium (Maristenkolleg Mindelheim)
06/1997	Allgemeine Hochschulreife
04/1998 bis 04/2002	Studium der Pharmazie an der Friedrich-Alexander-Universität Erlangen
04/2002	Zweites pharmazeutisches Staatsexamen
05/2002 bis 10/2002	Pharmaziepraktikantin in der Krankenhausapotheke des Universitätsklinikums Erlangen
11/2002 bis 08/2003	Pharmaziepraktikantin in der Adler-Apotheke Erlangen
10/2003	Erteilung der Approbation als Apothekerin
11/2003 bis 04/1005	Angestellte Apothekerin in der Adler-Apotheke Erlangen
05/2005 bis 07/2006	wissenschaftliche Mitarbeiterin am Institut für pharmazeutische Chemie der Julius-Maximillians-Universität Würzburg, AK Prof. Dr. Knut Baumann
08/2006 bis 03/2009	wissenschaftliche Mitarbeiterin am Institut für pharmazeutische Chemie der technischen Universität Carolo-Wilhelmina zu Braunschweig, AK Prof. Dr. Knut Baumann